



Fondamenti di Informatica

AA 2018/2019

Eng. Ph.D. Michela Paolucci

DISIT Lab <http://www.disit.dinfo.unifi.it>

Department of Information Engineering, DINFO

University of Florence

Via S. Marta 3, 50139, Firenze, Italy

tel: +39-055-2758515, fax: +39-055-2758570

michela.paolucci@unifi.it



UNIVERSITÀ
DEGLI STUDI
FIRENZE

DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

DISIT
DISTRIBUTED SYSTEMS
AND INTERNET
TECHNOLOGIES LAB

DISIT Lab
<http://www.disit.dinfo.unifi.it>
Department of Information Engineering,
DINFO
University of Florence
Via S. Marta 3, 50139, Firenze, Italy



Orario del Corso e Ricevimento

Insegnamento: FONDAMENTI DI INFORMATICA (A-L) (FonDiInf(A-L))

Ingegneria FIRENZE - A.A. 2018/2019 - 2° periodo

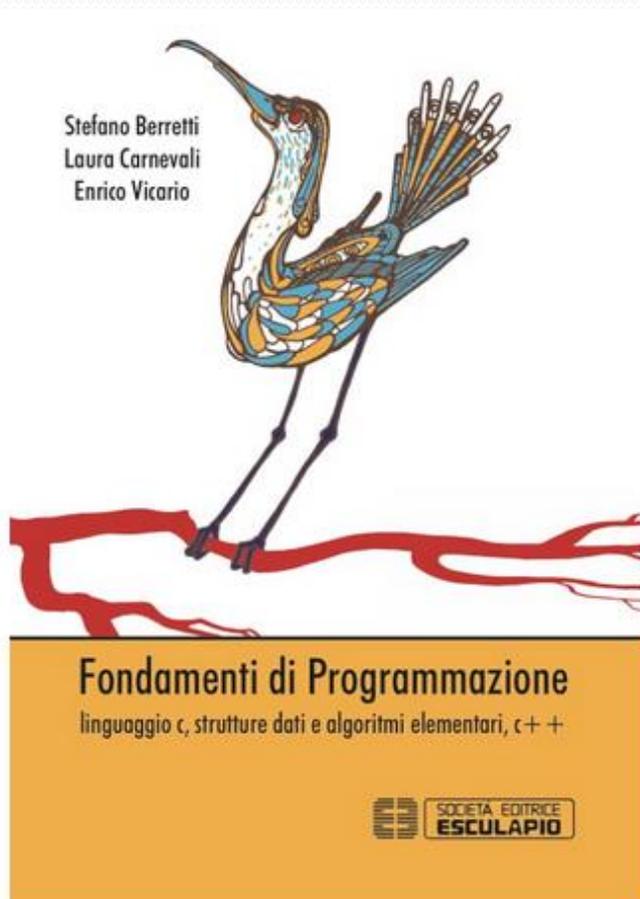
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
08:15						
09:15						
10:15						
11:15	(Auditorium A - C.D.M.)					
12:15	(Auditorium A - C.D.M.)					
14:00					(Auditorium A - C.D.M.)	
15:00					(Auditorium A - C.D.M.)	
16:00					(Auditorium A - C.D.M.)	
17:00						
18:00						

Docenti: PAOLUCCI MICHELA

Legenda: C.D.M.: Centro Didattico Morgagni

- Il ricevimento si svolge su appuntamento contattando la docente via email:
 - michela.paolucci@unifi.it

Libro di testo



- Stefano Berretti, Laura Carnevali, Enrico Vicario
- Fondamenti di programmazione. Linguaggio C, strutture dati, algoritmi elementari, C++
- Editore: Esculapio
- ISBN: 9788893850513
- **NOTA: Queste slide integrano e non sostituiscono quanto scritto sul libro di testo.**

Pagina del Corso

<http://www.disit.org/drupal/?q=node/7020>

Qui trovate:

- AVVISI
- Slide del corso
- Approfondimenti



Modalità esame:

- L'esame si compone di una prova scritta e una orale.
- La prova scritta consiste in alcuni elaborati di programmazione e nella eventuale discussione di contenuti del programma.
- La prova scritta è svolta su carta A4. Al termine, viene presentata e discussa la soluzione. Successivamente i candidati ricevono la fotocopia del loro elaborato.
- Per accedere alla prova orale, il candidato deve correggere il proprio elaborato, riportando le correzioni/variazioni in maniera visibile (i.e. uso di penna rossa) sulla fotocopia. Il candidato deve anche realizzare il programma corretto e funzionante che corregge l'elaborato e lo completa facendone un programma autocontenuto.
- Il candidato deve infine fornire una autovalutazione del proprio elaborato, in base al valore attribuito a ciascuna parte della prova, alla discussione della soluzione, all'esperienza acquisita nella correzione e realizzazione effettiva del programma.

Modalità di svolgimento:

- La prova orale può essere sostenuta a partire dalla settimana seguente alla prova scritta, non oltre la prova scritta successiva.
- La prova orale inizia con la discussione dell'elaborato, e prosegue eventualmente con l'approfondimento di tutti i contenuti del programma.

Esercizio 1 (1)

- Data una lista di valori interi in forma collegata con puntatori, scrivere una funzione che:
 - a. Calcoli la media dei valori.

Esercizio 1 (2)

- Data una lista di valori interi in forma collegata con puntatori, scrivere una funzione che:
 - a. Calcoli la media dei valori.
- Algoritmo:
 - Si **visita** la lista e si sommano i valori in essa contenuti (sum)
 - Si incrementa un contatore per contare gli elementi nella lista
 - Si ritorna la media, ovvero il rapporto fra la somma ed il numero di elementi calcolati

Esercizio 1 (3)

- Stiamo lavorando con una lista di valori interi in forma collegata con puntatori, quindi la lista risulta:

```
struct list {  
    int value;  
    struct list * next_ptr;  
};
```

- L'algoritmo per visita:

```
void visit(struct list * ptr) {  
    while (ptr != NULL) {  
        printf("%d\n", ptr->value);  
        ptr = ptr->next_ptr;  
    }  
}
```

Esercizio 1 (4)

- QUINDI:
 - Al posto della **stampa** del valore nella funzione del calcolo della media, ogni volta che si visita un elemento è necessario:
 - incrementare un contatore (numeroElementi) per contare gli elementi nella lista
 - Calcolare la somma dei valori visitati (sum)
 - Come risultato finale, si ritorna la media, ovvero il rapporto fra la somma ed il numero di elementi calcolati:
 - $media = sum / numeroElementi$

Esercizio 1 (5)

```
//dichiarazione della funzione
float meanList(struct list * ptr);
Main(){...}
//definizione della funzione
float meanList(struct list * ptr) {
    float sum = 0; //variabile di tipo float per il calcolo della media
    int numElem = 0; //variabile di tipo int per il numero di elementi
    while (ptr != NULL) { // visito gli elementi
        sum += ptr->value; //incremento sum
        numElem++; //incremento numero di elementi
        ptr = ptr->next_ptr; //aggiorno il puntatore al next
    }
    return sum / numElem; //rendo la media (valore di tipo float)
}
```

Esercizio 1 (6)

- Per verificare che la funzione scritta sia corretta è necessario chiamare la funzione da main. Questo comprende almeno le seguenti operazioni:
 - Inizializzare la lista
 - Inserire elementi nella lista
 - Richiamare la funzione appena creata (meanList) per effettuare il calcolo della media dei valori contenuti nella lista creata

```
#include <stdio.h> #include <stdlib.h>
```

```
#include <string.h>
```

Esercizio 1 (7)

```
struct list {
```

```
    int value;
```

```
    struct list * next_ptr;
```

```
};
```

```
void init(struct list ** ptrptr);
```

```
void pre_insert(struct list ** ptrptr, int value); //uso inserimento in testa
```

```
float meanList(struct list * ptr);
```

```
void getvalue(float *value_ptr);
```

```
void notify_selection_failure(char selection);
```

```
main(){
```

```
    struct list *lista;
```

```
    int size, value;
```

```
    size = 5;
```

```
    char selection[10];
```

```
    Boolean exit_required = FALSE;
```

```
    int inserimento = 0;
```

```
    int numeroElem =0;
```

```
    float sum = 0;
```

```
    init(&lista);  
    ..... }
```



```
... main(){
```

```
struct list *lista;
```

```
... init(&lista);
```

```
printf("Digita uno dei seguenti caratteri:\n");
```

```
printf("-A per avere la media degli elementi nella lista\n");
```

```
printf("-a per fare un inserimento in coda\n");
```

```
printf("-x per uscire dal programma\n");
```

```
do(
```

```
    printf("\nChe operazione vuoi fare? ");
```

```
    scanf("%s", selection);
```

```
    switch (selection[0]) {
```

```
        case 'A': //Calcolo del valore medio degli elementi nella lista
```

```
            sum = meanList(lista);
```

```
            printf("Ecco la media: %f", sum);
```

```
            break;
```

```
        case 'a':
```

```
            getvalue(&value); //inserimento in coda
```

```
            suf_insert(&lista, value);
```

```
            break; ...
```

```
    }
```

```
) while (exit_required == FALSE);
```

Esercizio 1 (8)



```
... main(){ ...
```

```
do(
```

Esercizio 1 (9)

```
printf("\nChe operazione vuoi fare? ");
```

```
scanf("%s", selection);
```

```
switch (selection[0]) {
```

```
    case 'A': //Calcolo del valore medio degli elementi nella lista
```

```
        sum = meanList(lista);
```

```
        printf("Ecco la media: %f", sum);
```

```
        break;
```

```
    case 'a':
```

```
        getvalue(&value); //inserimento in coda
```

```
        suf_insert(&lista, value);
```

```
        break; ...
```

```
    case 'x':
```

```
        exit_required = TRUE;
```

```
        break;
```

```
    default:
```

```
        notify_selection_failure(selection[0]);
```

```
}
```

```
) while (exit_required == FALSE);
```



Esercizio 1 (10)

```
//lista collegata con puntatori  
void init(struct list ** ptrptr) {  
    *ptrptr = NULL;  
}
```

```
//inserimento in coda
```

```
void suf_insert(struct list ** ptrptr, int value) {  
    /* inserimento in coda. Scorre il doppio puntatore  
    fino a che questo punta un puntatore nullo che,  
    per costruzione è il campo next_ptr dell'ultimo  
    elemento della lista, ovvero il puntatore su cui  
    applicare la malloc. Poi applica un inserimento  
    in testa sulla sottolista puntata dal doppio puntatore */  
    while (*ptrptr != NULL) {  
        ptrptr = &((*ptrptr)->next_ptr);  
    }  
    pre_insert(ptrptr, value);  
}
```

```
}
```

Esercizio 1 (11)

```
float meanList(struct list * ptr) {  
    float sum = 0;  
    int numElem = 0;  
    while (ptr != NULL) {  
        sum += ptr->value;  
        numElem++;  
        ptr = ptr->next_ptr;  
    }  
    return sum / numElem;  
}
```

Esercizio 1 (12)

```
void getvalue(int *value_ptr) {  
    //acquisisce un float da tastiera (passaggio per puntatore)  
    printf("Inserisci un valore: ");  
    scanf("%d", value_ptr);  
}
```

```
void notify_selection_failure(char selection) {  
    //notifica il fallimento della selezione  
    printf("\n%c Selezione non legale!!", selection);  
}
```

Esercizio 1 – Variante nella chiamata della funzione meanList(...) (1)

- Supponiamo di voler restituire (su Console) la media dei valori in un array SE esistono valori nella lista
- Un messaggio di errore su (su Console) un messaggio di errore altrimenti
- Si crea una funzione per il calcolo del numero di elementi in una lista
- Si chiama da main tale funzione per la gestione dell'errore

Esercizio 1 - variante (2)

```
... main(){
struct list *lista;
... init(&lista);
printf("Digita uno dei seguenti caratteri:\n");
printf("-A per avere la media degli elementi nella lista\n");
printf("-a per fare un inserimento in testa\n");
printf("-x per uscire dal programma\n");
do(
    printf("\nChe operazione vuoi fare? ");
    scanf("%s", selection);
    switch (selection[0]) {
        case 'A'://Calcolo del valore medio degli elementi nella lista
            sum = meanList(lista);
            numeroElem = numeroElementi(lista);
            if (numeroElem==0)
                printf("Non ci sono elementi nella lista");
            else
                printf("Ecco la media: %f", sum);
            break;
    }
) while (exit required == FALSE);
```

Esercizio 1 – Variante nella chiamata della funzione meanList(...) (2)

```
int numeroElementi(struct list * ptr); //dichiarazione
```

```
int numeroElementi(struct list * ptr){//definizione
    int numElem = 0;
    while (ptr != NULL) {
        numElem++;
        ptr = ptr->next_ptr;
    }
    return numElem;
}
```

Esercizio 2 (1)

- Data una lista di valori interi in forma collegata con puntatori, scrivere una funzione che:
 - Calcoli la deviazione standard dei valori

Esercizio 2 (2)

- Data una lista di valori interi in forma collegata con puntatori, scrivere una funzione che:
 - Calcoli la deviazione standard dei valori
- Algoritmo:
 - Si **visita** la lista e per ogni elemento:
 - Si incrementa un contatore (numeroElementi) per contare gli elementi nella lista
 - Si incrementa il valore *sum pari a*: $(x_i - \bar{x})^2$
 - Infine si ritorna la deviazione standard dei valori totali:

$$dev = \frac{1}{N} * sum = \frac{1}{N} \sqrt{\sum_{i=0}^N (x_i - \bar{x})^2}$$

con N = numero elementi e \bar{x} = media valori lista

Esercizio 2 (3)

- Stiamo lavorando con una lista di valori interi in forma collegata con puntatori, quindi la lista risulta:

```
struct list {  
    int value;  
    struct list * next_ptr;  
};
```

- L'algoritmo per visita:

```
void visit(struct list * ptr) {  
    while (ptr != NULL) {  
        printf("%d\n", ptr->value);  
        ptr = ptr->next_ptr;  
    }  
}
```

Esercizio 2 (4)

- QUINDI:
 - Al posto della **stampa** del valore nella funzione del calcolo della media, ogni volta che si visita un elemento è necessario:
 - Si incrementa un contatore (numeroElementi) per contare gli elementi nella lista
 - Si incrementa il valore *sum pari a*: $(x_i - \bar{x})^2$

$$dev = \frac{1}{N} * sum = \frac{1}{N} \sqrt{\sum_{i=0}^N (x_i - \bar{x})^2}$$

*con N = numero elementi e \bar{x}
= media valori lista (calcolata prima)*

Esercizio 2 (5)

- **NOTA:**
 - Questo quesito è collegato al precedente:
 - si sfrutta allora la funzione `meanList(...)` per il calcolo della media dei valori di una lista (di valori interi in forma collegata con puntatori)
- Si crea la funzione `stdList(...)` che calcola la deviazione standard **RICHIAMANDO** la funzione `meanList(...)`

Esercizio 2 (7)

```
float stdList(struct list * ptr){
    float mean = meanList(ptr);
    float sum = 0;
    int numElem = 0;
    while(ptr!=NULL){
        sum += (ptr->value-mean)*(ptr->value-mean);
        numElem++;
        ptr=ptr->next_ptr;
    }
    return sqrt(sum/numElem);
}
//NOTA: inclusione della libreria <math.h>
```

$$dev = \frac{1}{N} * sum = \frac{1}{N} \sqrt{\sum_{i=0}^N (xi - \bar{x})^2}$$

*con N = numero elementi e \bar{x}
= media valori lista (calcolata prima)*

Esercizio 2 (8)

- Si può aggiungere la funzionalità del calcolo della varianza al programma visto prima:

Si ricorda:

...

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
... struct list {
```

```
int value;
```

```
struct list * next_ptr;
```

```
};
```

...

```
float meanList(struct list * ptr); //funzione per il calcolo della media
```

```
int numeroElementi(struct list * ptr);
```

```
float stdList(struct list * ptr);
```

...

```
...
main(){//struct list *lista; Esercizio 2 (9)
```

```
...
float dev=0;
printf("\nChe operazione vuoi fare? ");
scanf("%s", selection);
do{
    switch (selection[0]) {
        ...
        case 'B':
            //Calcolo deviazione standard
            dev = stdList(lista);
            numeroElem = numeroElementi(lista);
            if (numeroElem == 0)
                printf("Non ci sono elementi nella lista");
            else
                printf("Ecco la deviazione standard: %f", dev);
            break; ...
    } while (exit_required == FALSE);
```



Esercizio 3 (1)

- Dato un array statico a e uno dinamico b delle stesse dimensioni, scrivere una funzione che:
 - a. Ritorni un nuovo array contenente la somma per componenti di $a+b$
 - b. Ritorni il prodotto per componenti dei due array direttamente sull'array a .

Esercizio 3 (2)

- Si lavora su due array che hanno stesse dimensioni
- La funzione da creare riceve in ingresso:
 - I due array (A, B)
 - Il numero degli elementi degli array
- Per il punto a):
 - si alloca un nuovo **array (D) dinamicamente** e si inseriscono i valori della somma dei singoli elementi di A e B
 - $D[i] = A[i] + B[i]$
- Per il punto b)
 - si sovrascrivono in A i valori del prodotto $a*b$. Il vettore è già allocato fuori dalla funzione
- NOTA:
 - La memoria allocata dinamicamente rimane accessibile fuori dalla funzione mediante il puntatore ritornato

Allocazione dinamica di un array

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
main(){
    int i, N =10;
    int *D = (int *)calloc(N, sizeof(int));
    for (i = 1; i < N; i++){
        D[i] = 2*(i+1);
    }
    printf("Ecco i valori contenuti in D:\n");
    for (i = 0; i < N; i++){
        printf("%d\n", D[i]);
    }
}
```

The C library function **void *calloc(size_t nitems, size_t size)** allocates the requested memory and returns a pointer to it. The difference in **malloc** and **calloc** is that **malloc** does not set the memory to zero where as **calloc** sets allocated memory to zero.

```

Main.c Main.c Main.c
esercitazione2Array (Ambito globale)
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5 main(){
6     int i, N = 10;
7     int *D = (int *)malloc(N* sizeof(int));
8     /*
9     for (i = 1; i < N; i++){
10        D[i] = 2*(i+1);
11    }*/
12    printf("Ecco i valori contenuti in D:\n");
13    for (i = 0; i < N; i++){
14        printf("D[%d] = %d\n", i, D[i]);

```

Uso di malloc()

Variabili locali

Nome	Valore
D	0x00ce9e10 {-842150451}
	-842150451
i	-858993460
N	10



```
C:\WINDOWS\system32\cmd.exe
Ecco i valori contenuti in D:
D[0] = -842150451
D[1] = -842150451
D[2] = -33698355
D[3] = 4521469
D[4] = -10230793
D[5] = -2147480767
D[6] = 6225998
D[7] = 5177426
D[8] = 5046337
D[9] = 5111881
Premere un tasto per continuare . . .
```

Uso di malloc()

- Uso di malloc() -> Array NON inizializzato

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4
5  main(){
6      int i, N =10;
7      int *D = (int *)calloc(N, sizeof(int));
8      /*
9      for (i = 1; i < N; i++){
10         D[i] = 2*(i+1);
11     }*/
12     printf("Ecco i valori contenuti in D:\n"); ≤ 1 ms trascorsi
13     for (i = 0; i < N; i++){
14         printf("D[%d] = %d\n", i, D[i]);
15     }
16 }
17

```

82 %

Variabili locali

Nome	Valore
D	0x01318530 {0}
i	-858993460
N	10

Uso di calloc()

```
C:\WINDOWS\system32\cmd.exe

Ecco i valori contenuti in D:
D[0] = 0
D[1] = 0
D[2] = 0
D[3] = 0
D[4] = 0
D[5] = 0
D[6] = 0
D[7] = 0
D[8] = 0
D[9] = 0
Premere un tasto per continuare . . .
```

Uso di calloc()

- Uso di calloc() -> Array inizializzato

Allocazione dinamica di un array

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
main(){
    int i, N =10;
    int *D = (int *)calloc(N, sizeof(int));
    for (i = 0; i < N; i++){
        D[i] = 2*(i+1);
    }
    printf("Ecco i valori contenuti in D:\n");
    for (i = 0; i < N; i++){
        printf("%d\n", D[i]);
    }
}
```

The C library function **void *calloc(size_t nitems, size_t size)** allocates the requested memory and returns a pointer to it. The difference in **malloc** and **calloc** is that **malloc** does not set the memory to zero where as **calloc** sets allocated memory to zero.

```
C:\WINDOWS\system32\cmd.exe
Ecco i valori contenuti in D:
D[0] = 2
D[1] = 4
D[2] = 6
D[3] = 8
D[4] = 10
D[5] = 12
D[6] = 14
D[7] = 16
D[8] = 18
D[9] = 20
Premere un tasto per continuare . . .
```

- Uso di `calloc()` -> Array inizializzato con i primi N numeri pari

...tornando all'esercizio 3



Esercizio 3 (1)

- Dato un array statico a e uno dinamico b delle stesse dimensioni, scrivere una funzione che:
 - a. Ritorni un nuovo array contenente la somma per componenti di $a+b$
 - b. Ritorni il prodotto per componenti dei due array direttamente sull'array a .

Esercizio 3 (2)

- Si lavora su due array che hanno stesse dimensioni
- La funzione da creare riceve in ingresso:
 - I due array (A, B)
 - Il numero degli elementi degli array
- Per il punto **a)**:
 - si alloca un nuovo **array (D) dinamicamente** e si inseriscono i valori della somma dei singoli elementi di A e B
 - $D[i] = A[i] + B[i]$
- Per il punto **b)**:
 - si sovrascrivono in **A** i valori del prodotto dei singoli elementi di A per B:
 - $A[i] = A[i] * B[i]$
 - Il vettore **A** DEVE essere già allocato fuori dalla funzione
- **NOTA:**
 - **La memoria allocata dinamicamente rimane accessibile fuori dalla funzione mediante il puntatore ritornato**

Esercizio 3 (3)

```
int *arrayfunction(int* A, int* B, int N); //dichiarazione
```

```
int *arrayfunction(int* A, int* B, int N) { //definizione
    int *D = (int *)calloc(N, sizeof(int));
    for (int i = 0; i < N; i++) { // Punto a)
        D[i]=A[i]+B[i]; //nuovo array D contenente la
            //somma per componenti di a+b
    }
    for ( int i = 0; i < N; i++ ){ // Punto b) Ritorna il prodotto per
//componenti dei due array direttamente sull'array A,
        A[i]=A[i]*B[i];
    }
    return D;
}
```

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

Esercizio 3 (4)

```
int *arrayfunction(int* A, int* B, int N);
```

```
main(){
```

```
    int i, N =5;
```

```
    int *A = (int *)calloc(N, sizeof(int)); //array allocto in modo dinamico
```

```
    int B[] = {23, 5, 0, 67, 89}; //array allocato in modo statico e inizializzato
```

```
    int *E = NULL;
```

```
    for (i = 0; i < N; i++){//inizializzo A con i primi N numeri pari
```

```
        A[i] = 2*(i+1);
```

```
    }
```

```
    printf("Ecco i valori iniziali in A:\n");
```

```
    for (i = 0; i < N; i++) {
```

```
        printf("A[%d] = %d\n", i, A[i]);
```

```
    }
```

```
    printf("Ecco i valori iniziali in B:\n");
```

```
        for (i = 0; i < N; i++) {
```

```
            printf("B[%d] = %d\n", i, B[i]);
```

```
    ...
```



```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

Esercizio 3 (5)

```
int *arrayfunction(int* A, int* B, int N);
main(){
```

```
    ...
    E = arrayfunction(A, B, N);
    printf("Punto a) Ecco i valori finali contenuti in E.\n");
    printf("Ovvero, E contiene la somma dei singoli valori degli elementi di
           A e B:\n");
    for (i = 0; i < N; i++) {
        printf("E[%d] = %d\n", i, E[i]);
    }
    printf("Punto b) Ecco i valori finali contenuti in A:\n");
    for (i = 0; i < N; i++){
        printf("A[%d] = %d\n", i, A[i]);
    }
}
```

Ecco i valori iniziali in A:

A[0] = 2

A[1] = 4

A[2] = 6

A[3] = 8

A[4] = 10

Ecco i valori iniziali in B:

B[0] = 23

B[1] = 5

B[2] = 0

B[3] = 67

B[4] = 89

Punto a) Ecco i valori finali contenuti in E.

Ovvero, E contiene la somma dei singoli valori degli elementi di A e B:

E[0] = 25

E[1] = 9

E[2] = 6

E[3] = 75

E[4] = 99

Punto b) Ecco i valori finali contenuti in A:

A[0] = 46

A[1] = 20

A[2] = 0

A[3] = 536

A[4] = 890

Premere un tasto per continuare . . .

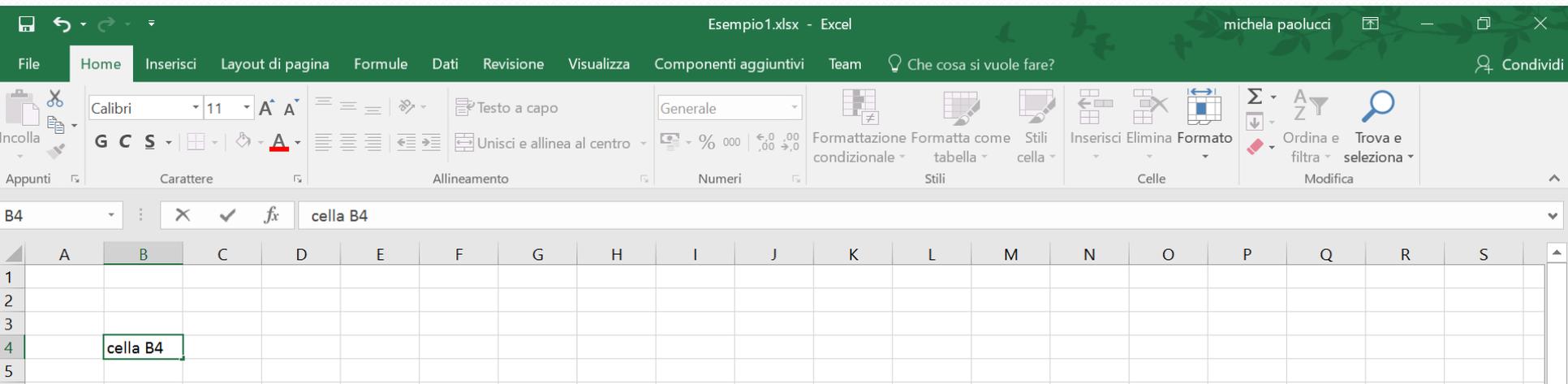
Excel



Outline

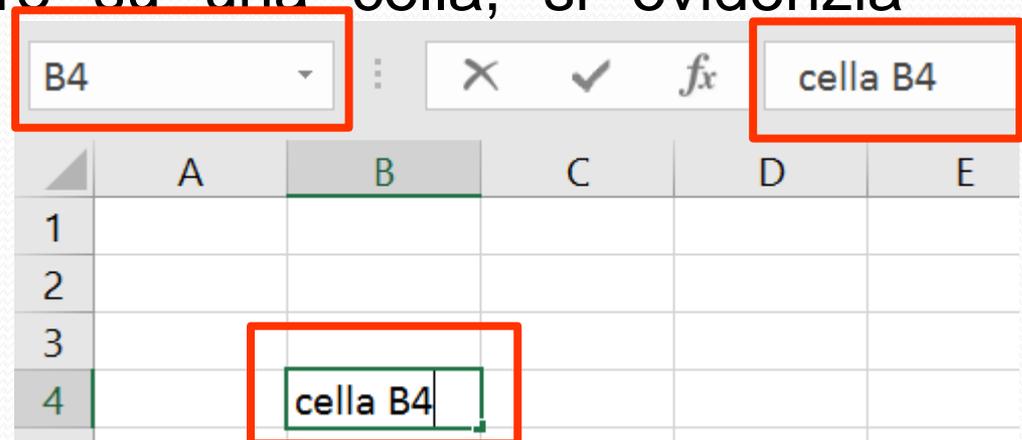
- Excel 
 - Operazioni/formule di base
 - Filtri, Somma, Media, Max e Min, creazione formule semplici, etc.
 - Analisi dati: Tabelle e grafici
 - Macro

Introduzione (1)



Celle (1)

- Ogni foglio di Excel è diviso in righe e colonne che hanno un nome:
 - il nome delle righe corrisponde al numero della riga sul foglio
 - Il nome delle colonne ad una lettera alfabetica (quando sono finite le lettere, si usano due caratteri invece di uno, per esempio AX)
- I nomi delle righe e delle colonne sono detti **INTESTAZIONI**
- L'intersezione tra una riga e una colonna è detto **CELLA**
- Se si decide di lavorare su una cella, si evidenzia rispetto alle altre



Celle (2)

- Per spostarsi tra le celle si usa il tasto 'tabulazione' (il cursore si sposta di una cella a destra)
- Ogni cella può contenere diversi tipi di informazioni:
 - Date, numeri, testo, etc.
- E' possibile impostare il formato di ogni cella (Selezionare cella > tasto destro > Formato celle), ovvero decidere il modo in cui visualizzare un dato:
 - Come scrivere una data (14/03/2012 o 14/03, ...)
 - Bordo di una cella
 - Tipo di carattere
 - Etc.

Esempio1.xlsx - Excel

File Home Inserisci Layout di pagina Formule Dati Revisione Visualizza Componenti aggiuntivi Team ?

Calibri 11 A A Testo a capo Generale

G C S Unisci e allinea al centro % 000 0,00 0,00 Formattazione condizionale

Appunti Carattere Allineamento Numeri

B4 cella B4

	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3											
4		cella B4									
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											

Foglio1

Pronto

Formato celle

Numero Allineamento Carattere Bordo Riempimento Protezione

Categoria:

- Generale
- Numero
- Valuta
- Contabilità
- Data**
- Ora
- Percentuale
- Frazione
- Scientifico
- Testo
- Speciale
- Personalizzato

Esempio
cella B4

Tipo:

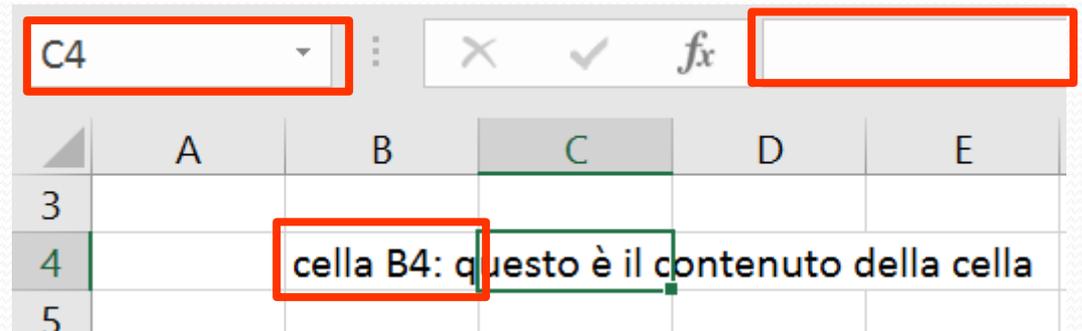
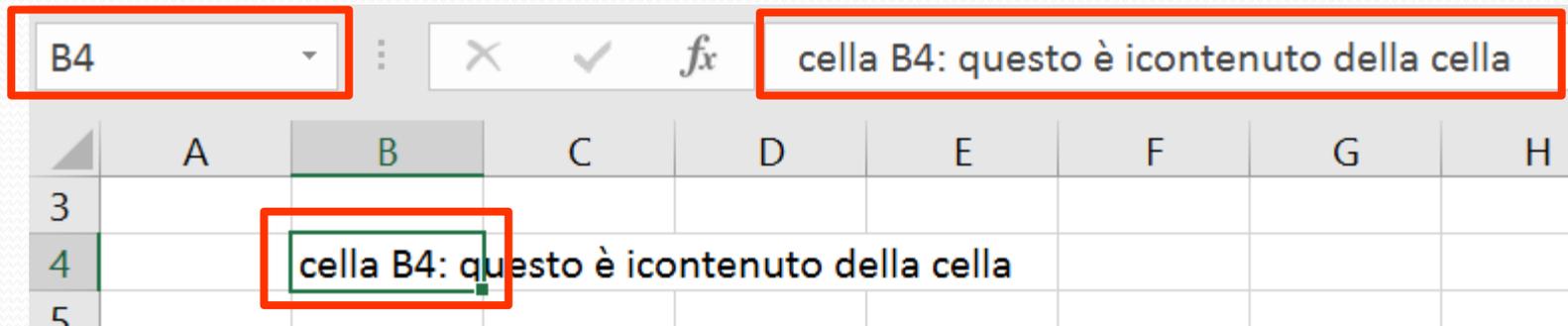
- *14/03/2012**
- *mercoledì 14 marzo 2012
- 14/3
- 14/3/12
- 14/03/12
- 14-mar
- 14-mar-12

Impostazioni locali (località):
Italiano (Italia)

I formati di data visualizzano i numeri seriali di data e ora come valori di data. I formati di data che iniziano con un asterisco (*) riflettono le modifiche apportate alle impostazioni internazionali di data e ora specificate per il sistema operativo. I formati senza asterisco sono invece indipendenti dalle impostazioni del sistema operativo.

Celle (3)

- Dimensione delle celle
- Le celle hanno una dimensione standard, se un testo 'esce' da tale dimensione e la cella alla sua destra è libera, allora excel lo visualizza per intero MA è sempre contenuto in una unica cella (quella in cui è stato inserito)



Selezione celle (1)

- E' possibile selezionare più celle insieme (Shift + frecce) per effettuare sulle celle selezionate delle operazioni (anche solo di visualizzazione, ad esempio mettere un bordo, etc.)

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I
3									
4		cella B4: questo è il contenuto della cella							
5									
6		altro dato							
7					23578				
8									
9									
10									
11									
12									
13									
14									
15									
16									

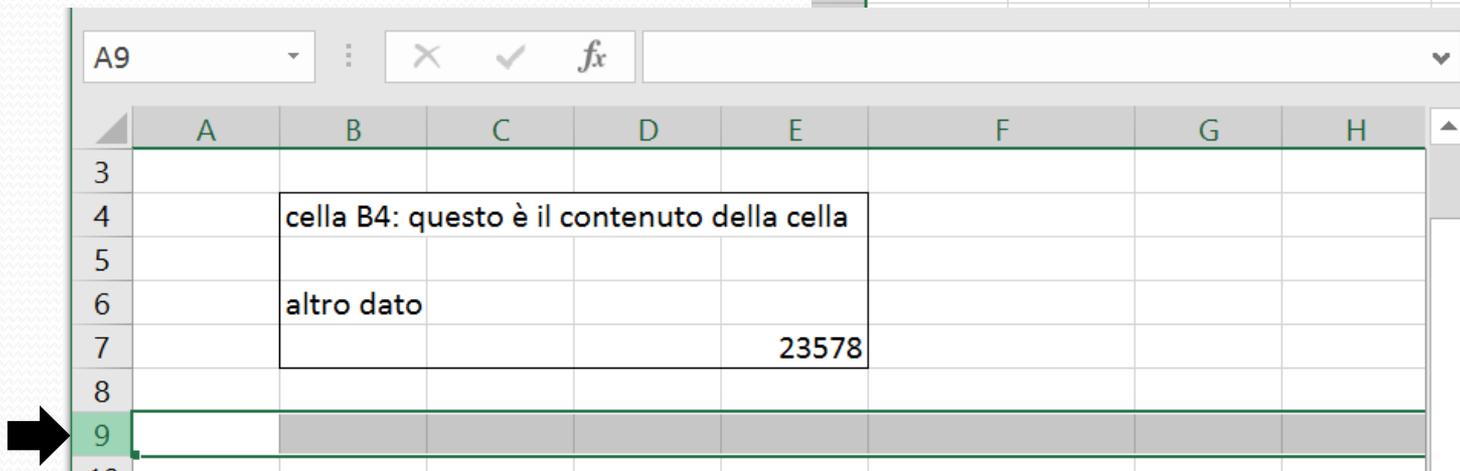
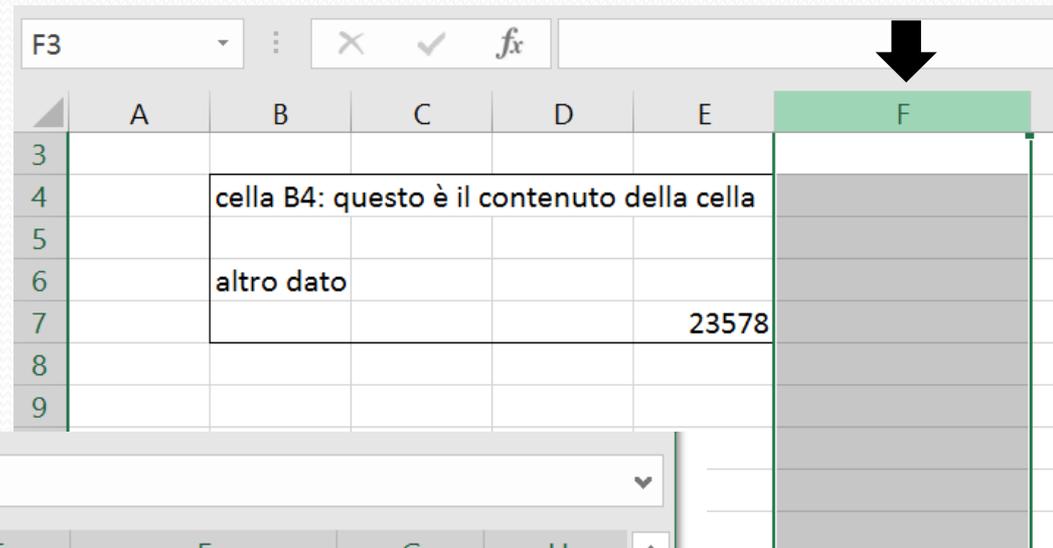
The context menu for the selection B4:E7 includes the following options:

- Formattazione (selected)
- Grafici
- Totali
- Tabelle
- Grafici sparkline
- Barre dei dati
- Scala di colori
- Set di icone
- Maggiore di
- Primo 10%
- Cancella formatta...

La formattazione condizionale usa regole per evidenziare i dati interessanti.

Selezione celle (2)

- E' possibile eseguire operazioni su colonne o righe:
 - Selezione
 - Inserimento
 - Cancellazione
 - Etc.



Fogli di lavoro

- Un file excel può contenere più fogli di lavoro
- Copia
- Aggiunta nuovo
- Etc.

The screenshot shows an Excel spreadsheet with the following content:

	A	B	C	D	E	F	G	H
3								
4		cella B4: questo è il contenuto della cella						
5								
6		altro dato						
7					23578			
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

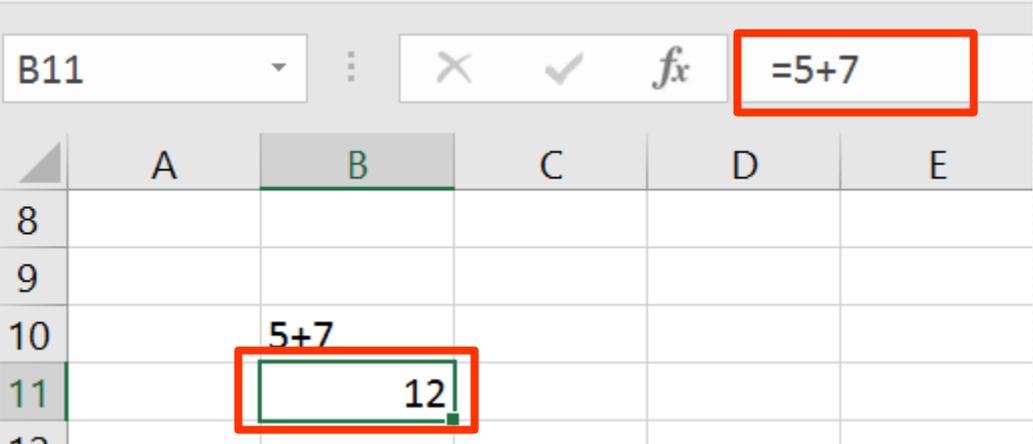
The dialog box "Sposta o copia" is open, showing the following options:

- Sposta i fogli selezionati
- Nella cartella: Esempio1.xlsx
- Prima del foglio: Foglio1 (sposta alla fine)
- Crea una copia

Buttons: OK, Annulla

Formule (1)

- Uno degli usi dei fogli di calcolo è quello di fornire il risultato di alcune operazioni
- Se si vuole eseguire la somma tra due numeri:
 - Digitare '=5+7' (senza spazi) e cliccare invio
- Si possono usare le varie operazioni:
 - Somma (+)
 - Sottrazione (-)
 - Moltiplicazione (*)
 - Divisione (/)
 - Elevamento a potenza (^)

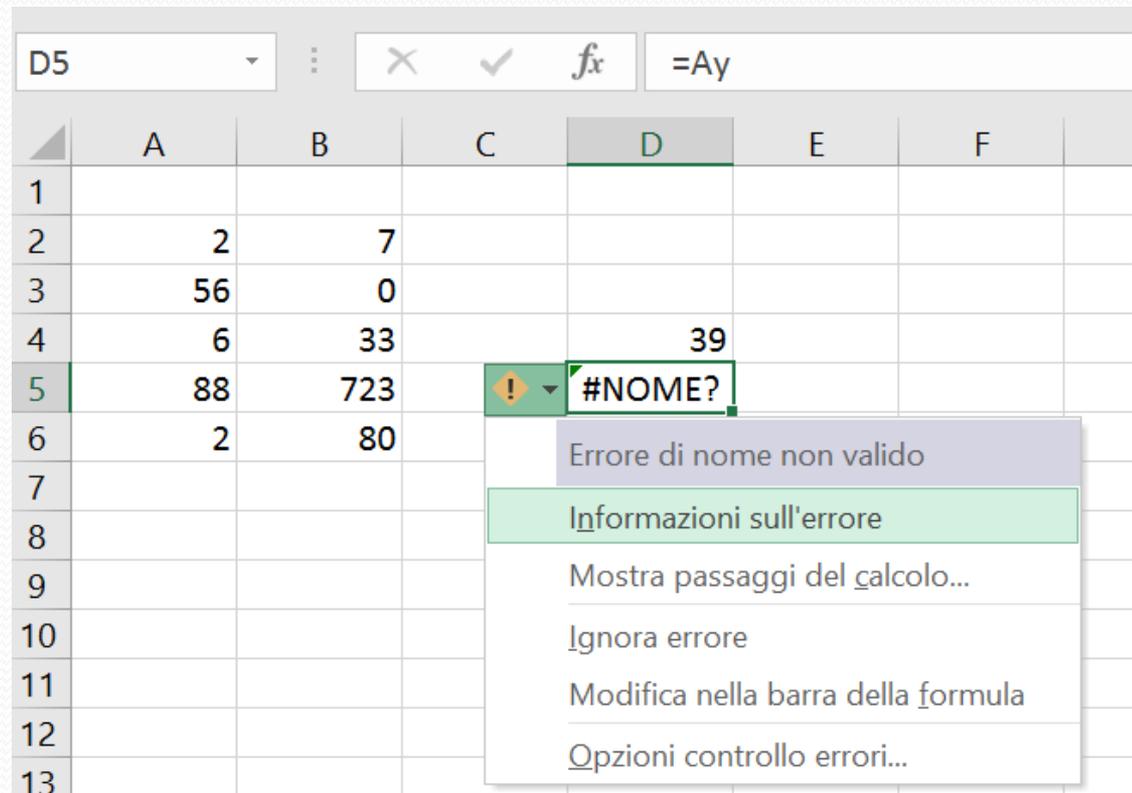


The screenshot shows a spreadsheet interface. At the top, the formula bar displays the formula `=5+7`, which is highlighted with a red box. Below the formula bar, the spreadsheet grid is visible. The columns are labeled A, B, C, D, and E. The rows are labeled 8, 9, 10, 11, and 12. Cell B10 contains the text `5+7`. Cell B11 contains the result `12`, which is also highlighted with a red box.

	A	B	C	D	E
8					
9					
10		5+7			
11		12			
12					

Formule (3)

- Se si scrive in modo errato una formula, nella cella corrispondente si trova la stringa '#NOME?' e una serie di possibili indicazioni relative al possibile tipo di errore effettuato



	A	B	C	D	E	F
1						
2	2	7				
3	56	0				
4	6	33		39		
5	88	723		#NOME?		
6	2	80				
7						
8						
9						
10						
11						
12						
13						

Formule (4)

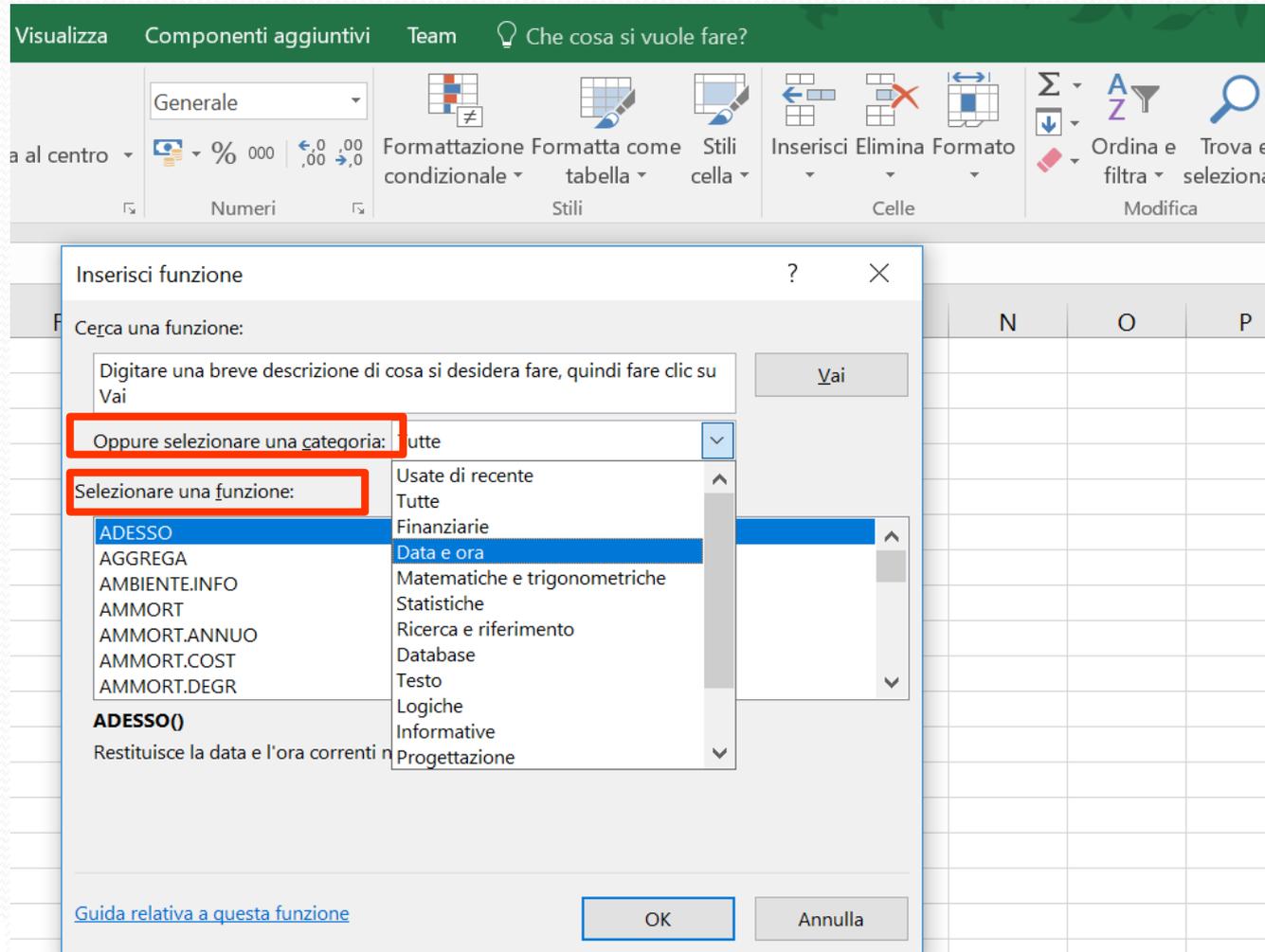
- E' possibile selezionare più celle in una colonna/riga ed effettuare delle operazioni

The screenshot shows the Microsoft Excel interface. The 'Inserisci funzione' (Insert Function) dialog box is open, displaying a list of functions. The 'SOMMA' (SUM) function is selected. Below the list, the formula **SOMMA(num1;num2;...)** is highlighted in a red box, with a red arrow pointing to the formula bar. The formula bar contains **=SOMMA(C2:C6)**, also highlighted in a red box. In the background, a spreadsheet is visible with columns B and C. Column B is labeled 'Colonna B' and column C is labeled 'ColonnaC'. The data in column B is 2, 56, 6, 88, 2. The data in column C is 7, 0, 33, 723, 80. The 'Somma' (Sum) row shows the total for column B as 154 and for column C as 843. A tooltip for the 'Somma' button is also visible, explaining that it automatically sums the selected data.

	Colonna B	ColonnaC
2	2	7
3	56	0
6	6	33
88	88	723
2	2	80
Somma	154	843

Funzioni (1)

- Excel prevede una serie di funzioni per la esecuzione di alcune operazioni
 - SOMMA
 - ARROTONDA
 - MEDIA
 - SEN
 - COS
 - Etc.



Funzioni (2)

The image shows a screenshot of Microsoft Excel with the 'Inserisci funzione' (Insert Function) dialog box open. The dialog box is titled 'Inserisci funzione' and has a search bar with the text 'Cerca una funzione:'. Below the search bar, there is a dropdown menu for 'Oppure selezionare una categoria:' set to 'Statistiche'. A list of functions is shown, with 'MEDIA' selected. Below the list, the function 'MEDIA(num1;num2;...)' is highlighted in a red box, with a description: 'Restituisce la media aritmetica degli argomenti (numeri, nomi o riferimenti contenenti numeri)'. A red arrow points from this box to a browser window on the right. The browser window shows the 'Sintassi' (Syntax) page for the 'MEDIA' function, with the URL 'https://support.office'. The page content includes the function name 'MEDIA (num1; [num2]; ...)' and a list of arguments: 'Num1' (obligatory) and 'Num2; ...' (optional). The browser window is also outlined in red.

File Home Inserisci Layout di pagina Formule Dati Revisione Visualizza Componenti aggiuntivi Team Che cosa si vuole fare?

Calibri 11 A A Testo a capo Generale

Unisci e allinea al centro % 000 ,00 ,00

Formattazione condizionale Formatta come tabella Stili cella

Inserisci Elimina Formato

Ordina e filtra Trova e seleziona

B9

A B C D E F G H I J K L M N O P Q

1 Inserisci funzione

2 Cerca una funzione:

3 Digita una breve descrizione di cosa si desidera fare, quindi fare clic su Vai

4 Oppure selezionare una categoria: Statistiche

5

6

7 Selezionare una funzione:

8 MAX.VALORI

9 MEDIA

10 MEDIA.ARMONICA

11 MEDIA.DEV

12 MEDIA.GEOMETRICA

13 MEDIA.PIÙ.SE

14 MEDIA.SE

15 **MEDIA(num1;num2;...)**

16 Restituisce la media aritmetica degli argomenti (numeri, nomi o riferimenti contenenti numeri).

17

18

19 [Guida relativa a questa funzione](#)

20 OK Annulla

21

22

MEDIA X Nuova scheda Nuova scheda Nuova scheda

https://support.office

Cerca

Sintassi

MEDIA (num1; [num2]; ...)

Gli argomenti della sintassi della funzione MEDIA sono i seguenti:

- **Num1** Obbligatorio. Primo numero, riferimento di cella o intervallo di cui si desidera calcolare la media.
- **Num2; ...** Facoltativo. Ulteriori numeri, riferimenti di cella o intervalli di cui si desidera calcolare la media. È possibile specificare fino a 255 valori.

Selezionare/Copiare/Trascinare celle

- Si possono selezionare celle e copiarle o tagliarle, incollandole in altre zone del foglio
- In alcuni casi predefiniti, il riempimento automatico compie azioni particolari
- Si selezionano una o più celle (in base alla operazione che si vuole fare), click tasto sinistro e si trascina

	A	B	C	D	E	F
1	17	Gennaio	Lunedì	lun	15/05/2017	
2	riempimento	Febbraio	Martedì	mar	16/05/2017	
3	riempimento	Marzo	Mercoledì	mer	17/05/2017	
4	riempimento	Aprile	Giovedì	gio	18/05/2017	
5	riempimento	Maggio	Venerdì	ven	19/05/2017	
6	riempimento	Giugno	Sabato	sab	20/05/2017	
7	riempimento	Luglio	Domenica	dom	21/05/2017	
8	riempimento	Agosto	Lunedì	lun		
9		Settembre	Martedì	mar		
10		Ottobre	Mercoledì	mer		
11		Novembre	Giovedì			
12		Dicembre	Venerdì			
13		Gennaio	Sabato			
14			Domenica			
15			Lunedì			
16			Martedì			
17			Mercoledì			
18			Giovedì			
19			Venerdì			
20			Sabato			
21			Domenica			

Riferimenti assoluti e relativi (1)

- E' possibile copiare anche le formule (sempre trascinandolo)

	A	B	C	D
1		Colonna B	ColonnaC	ColonnaD
2		2	7	9
3		56	0	
4		6	33	
5		88	723	
6		2	80	
7	Somma	154	843	

- Nelle celle D3-D6 excel memorizza la stessa formula contenuta in D2 per RIFERIMENTO RELATIVO

	A	B	C	D	E	F	G
1		Colonna B	ColonnaC	ColonnaD			
2		2	7	9			
3		56	0	56			
4		6	33	39			
5		88	723	811			
6		2	80	82			
7	Somma	154	843		0		

Opzioni riempimento automatico

Riferimenti assoluti e relativi (2)

- Per scrivere un riferimento ASSOLUTO si usa il carattere '\$'
- Formula:
 - B2+C2
- Formula con riferimento ASSOLUTO:
 - B2+\$C\$2
- Nelle celle D3-D6 excel memorizza la stessa formula contenuta in D2 per RIFERIMENTO ASSOLUTO

Formula bar: `=B2+C2`

	A	B	C	D	E
1		Colonna B	ColonnaC	ColonnaD	ColonnaE
2		2	7		=B2+\$C\$2
3		56	0	56	63
4		6	33	39	13
5		88	723	811	95
6		2	80	82	9
7	Somma	154	843		

Collegamenti tra fogli (1)

- Nelle formule è possibile fare riferimento alle celle di altri fogli di lavoro, con la sintassi seguente:
 - = NOMEFOGLIO ! NOMECELLA
- E' possibile anche comporre formule:
- = NOMEFOGLIO1 ! NOMECELLA1 + NOMEFOGLIO2 ! NOMECELLA2

	A	B	C
1		Colonna B	ColonnaC
2		2	7
3		56	0
4		6	33
5		88	723
6		2	80
7	Somma	154	843
8			
9		23578	
10			

Collegamenti tra fogli (2)

- Nelle formule è possibile fare riferimento alle celle di altri fogli di lavoro, con la sintassi seguente:
 - = NOMEFOGLIO ! NOMECELLA

	A	B	C
1		Colonna B	ColonnaC
2		2	7
3		56	0
4		6	33
5		88	723
6		2	80
7	Somma	154	843
8			
9		23578	
10			

Collegamenti tra fogli (3)

- Quando c'è bisogno di riferirsi a celle che si trovano in altri file, si devono indicare le seguenti informazioni:
 - Nome del file di origine;
 - Percorso completo del file: nome della directory in cui si trova il file, per esempio C:\DOCUMENTI
 - Nome del foglio di lavoro
 - Riferimento alle celle

='PERCORSO\DIRECTORY\[FILE.xls]FOGLIO'!RIFERIMENTO
CELLE

The screenshot illustrates the process of creating a cross-worksheet reference in Excel. The formula bar at the top shows the formula `=C:\Users\disit\Documents\Paolucci\FondamentiInformatica_2017\Slide\MacroCategorie\Excel\[Esempio2.xlsx]Foglio1!A1`. Below it, a smaller grid shows the source spreadsheet with cell A1 containing the value 34. The main grid shows the formula being entered into cell A1 of the current spreadsheet.

Collegamenti tra fogli (3)

- Quando c'è bisogno di riferirsi a celle che si trovano in altri file, si devono indicare le seguenti informazioni:
 - Nome del file di origine;
 - Percorso completo del file: nome della directory in cui si trova il file, per esempio C:\DOCUMENTI
 - Nome del foglio di lavoro
 - Riferimento alle celle

='PERCORSO\DIRECTORY\[FILE.xls]FOGLIO!'RIFERIMENTOCELLE

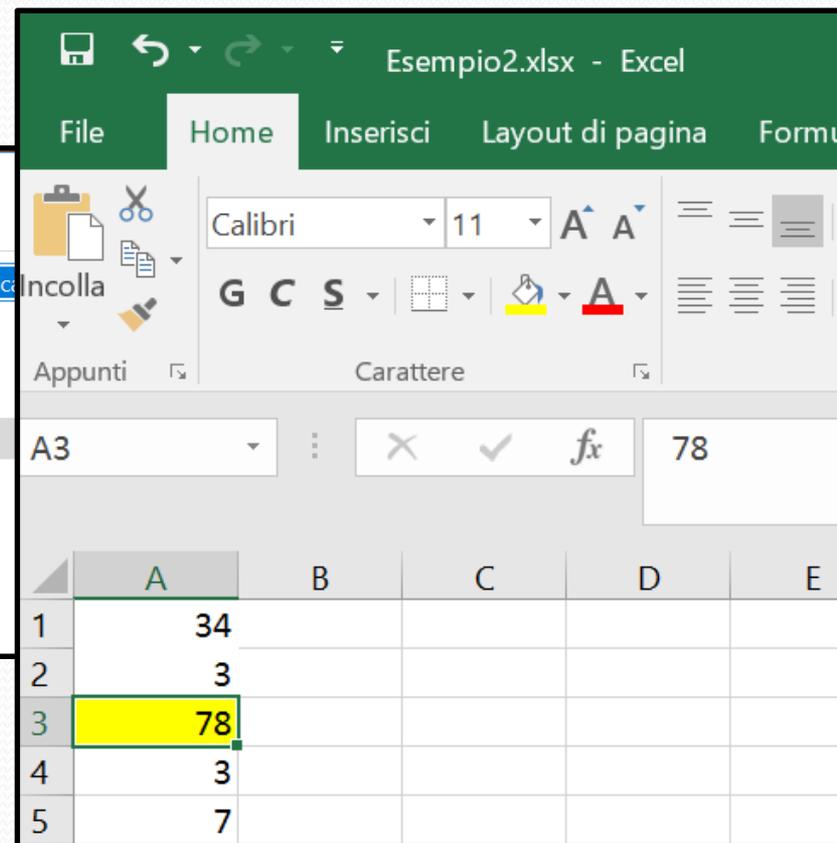
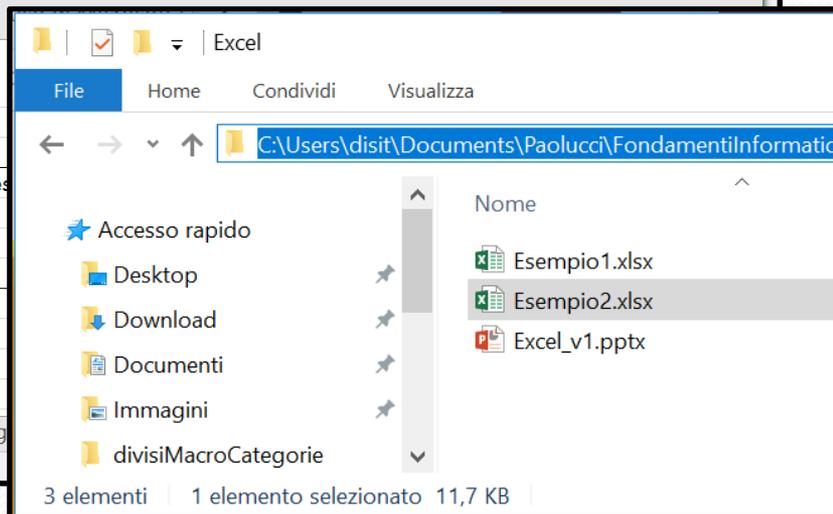
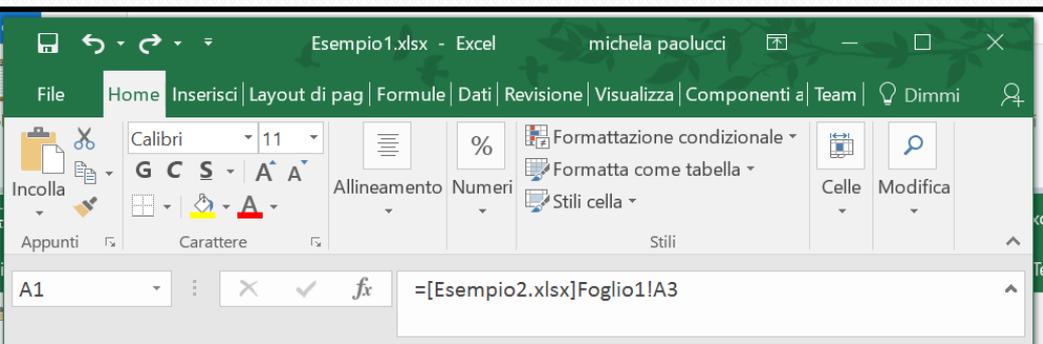
Esempio (cosa scrivere nella cella):

='C:\Users\disit\Documents\Paolucci\FondamentiInformatica_2019\Slide\MacroCategorie\Excel\[Esempio2.xlsx]Foglio1!'A1

The screenshot displays the Microsoft Excel interface. The formula bar at the top shows the formula: `=C:\Users\disit\Documents\Paolucci\FondamentiInformatica_2017\Slide\MacroCategorie\Excel\[Esempio2.xlsx]Foglio1!A3`. The spreadsheet grid below shows cell A1 containing the value 78. A smaller inset window shows a zoomed-in view of cell A1 with the formula bar containing `=Esempio2.xlsx]Foglio1!A3`.

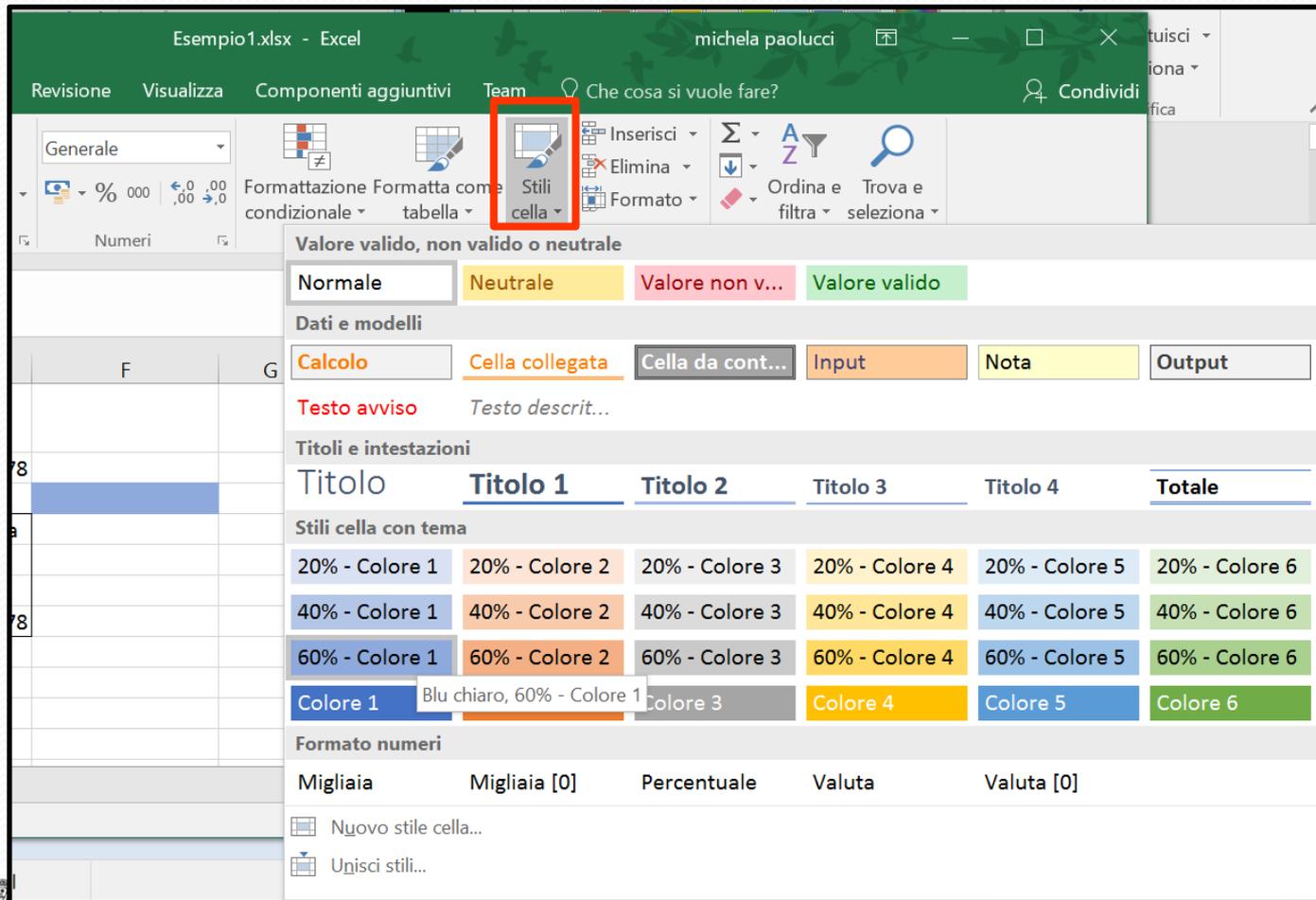
Collegamenti tra fogli (4)

- ='C:\Users\disit\Documents\Paolucci\FondamentiInformatica_2019\Slide\MacroCategorie\Excel\[Esempio2.xlsx]Foglio1!A3



Formattazione (1)

- Insieme di operazioni che riguardano la vista, ovvero la modalità con cui le informazioni vengono visualizzate



- Stile di una cella

Formattazione (2)

- Insieme di operazioni che riguardano la vista, ovvero la modalità con cui le informazioni vengono visualizzate

The screenshot shows the Microsoft Excel interface. The ribbon is set to 'Formattazione condizionale'. The 'Formatta come tabella' button is highlighted with a red box. A red arrow points from this button to a filter dropdown menu for column D. The filter menu shows options for sorting and filtering, with a search box containing '8' and a list of results including 82 and 811.

A	B	C	D	E	F	G
Colonna B	ColonnaC	ColonnaD				
2	7	9				
56	0	56				
6	33	39				
88	723	811				
2	80	82				

Stile di una tabella

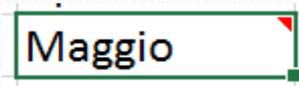
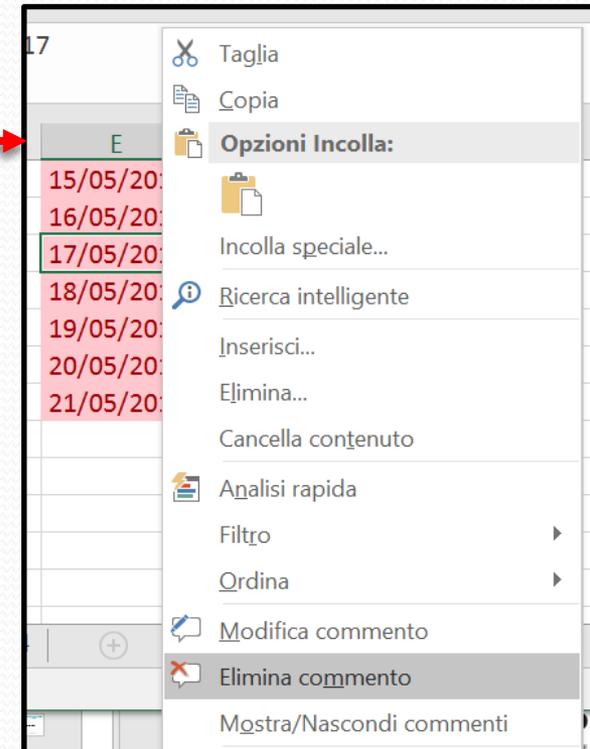
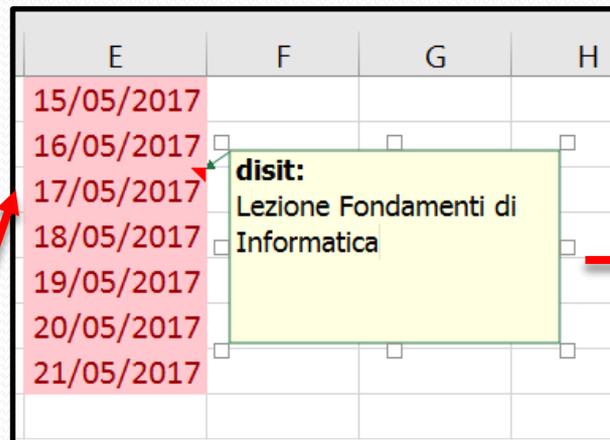
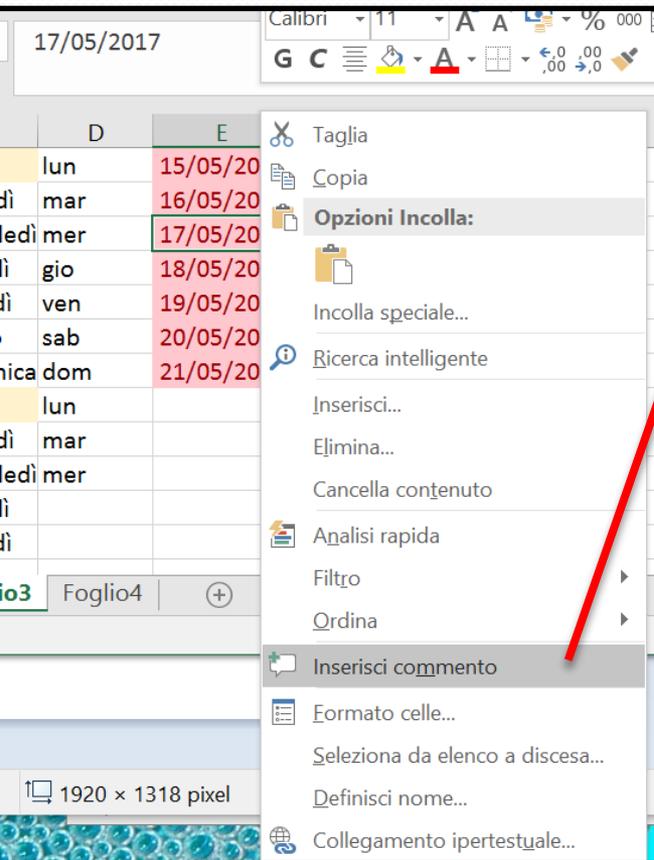
Formattazione (3)

The screenshot shows the Microsoft Excel interface with the 'Home' ribbon selected. The 'Formattazione condizionale' button is highlighted with a red box, and a red arrow points to a dialog box titled 'Data corrispondente a'. The dialog box is open, showing a list of date ranges with 'Questo mese' selected. The spreadsheet data shows dates from 15/05/2017 to 21/05/2017 in column E, with corresponding days of the week in columns C and D. The cells containing these dates are highlighted in light red.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	17	Gennaio	Lunedì	lun	15/05/2017									
2	riempimento	Febbraio	Martedì	mar	16/05/2017									
3	riempimento	Marzo	Mercoledì	mer	17/05/2017									
4	riempimento	Aprile	Giovedì	gio	18/05/2017									
5	riempimento	Maggio	Venerdì	ven	19/05/2017									
6	riempimento	Giugno	Sabato	sab	20/05/2017									
7	riempimento	Luglio	Domenica	dom	21/05/2017									
8	riempimento	Agosto	Lunedì	lun										
9		Settembre	Martedì	mar										
10		Ottobre	Mercoledì	mer										
11		Novembre	Giovedì											
12		Dicembre	Venerdì											

Commenti

- Selezione cella > Tasto destro > crea commento



- Cella con commento

Grafici (1)

- I grafici sono immagini che rappresentano i dati contenuti nelle tabelle
- In generale, risulta molto più semplice e immediato consultare un grafico che una tabella
- Esistono varie tipologie di grafici. La scelta del tipo di grafico da usare dipende:
 - Dai dati che abbiamo a disposizione
 - Dal tipo di informazione che vogliamo 'far emergere' tramite l'uso del grafico
- Alcuni dei grafici più usati sono:
 - Grafici a torta e istogrammi

Grafici (2)

- Creazione rapida di un grafico:
 - Selezione tabella > click strumento analisi rapida > grafici

	A	B	C	D	E	F	G	H	I
1		Italia	Francia	Grecia	Spagna				
2	Olive	1000	200	700	800				
3	Salumi	2000	500	100	1800				
4	formaggi	3000	4300	870	350				
5	Arance	500	200	170	600				
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									

Formattazione **Grafici** Totali Tabelle Grafici sparkline







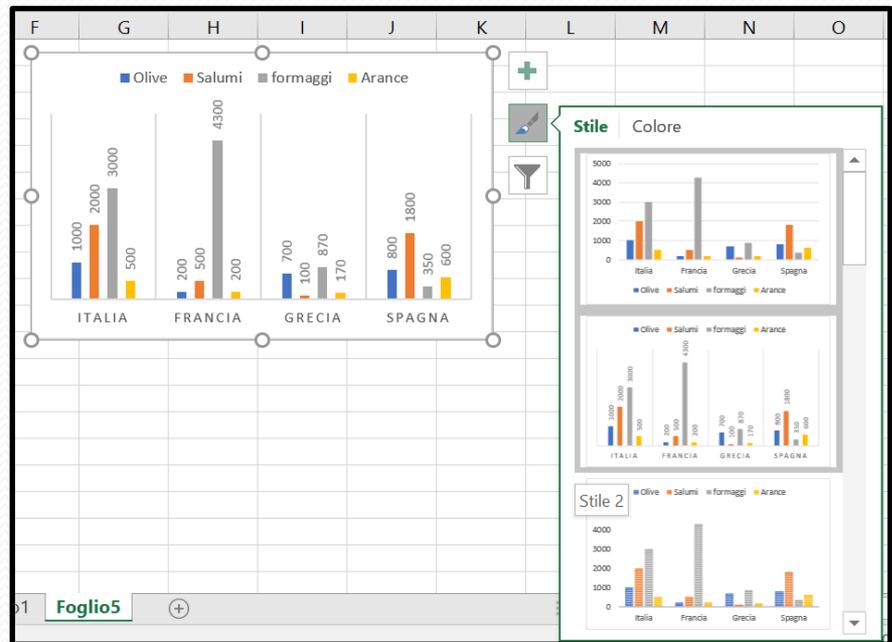
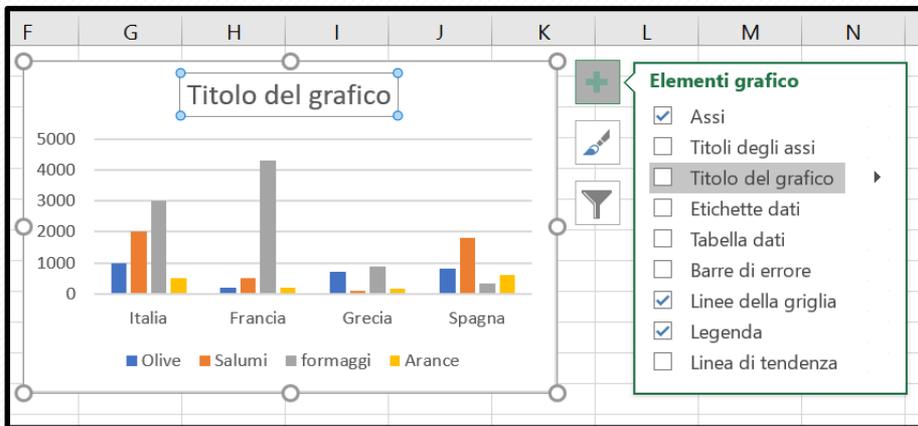

Colonne raggrup... Colonne raggrup... Colonne in pila Colonne in pila Barre raggrup... Altri grafici

I grafici consigliati semplificano la visualizzazione dei dati.



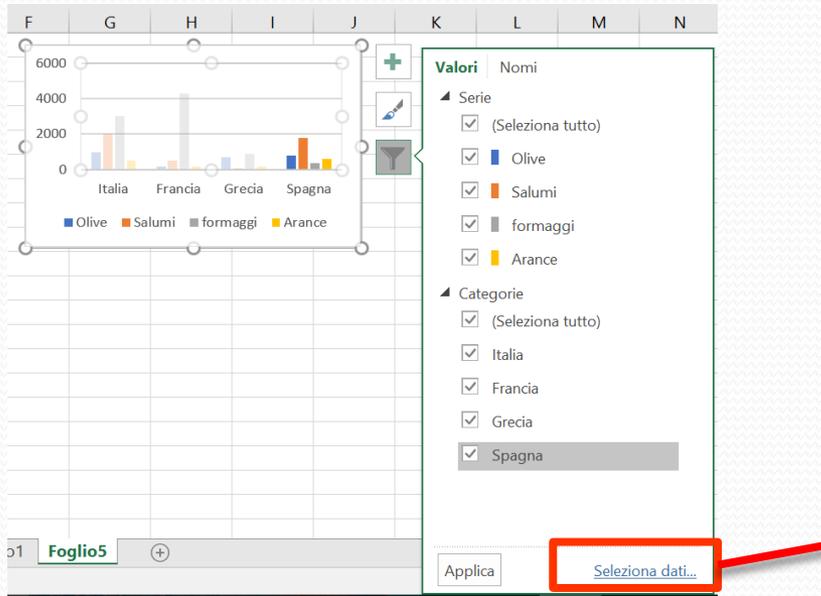
Grafici (3)

- Modifica del grafico:
 - Elementi aggiuntivi: legenda, etichette, titolo, etc.
 - Stile tabella



Grafici (4)

- Filtri e Modifica ai dati da visualizzare nel grafico



- Tabella: =Foglio5!\$A\$1:\$E\$5

The 'Seleziona origine dati' dialog box shows the data source as '=Foglio5!\$A\$1:\$E\$5'. It has a 'Scambia righe/colonne' button. Below, there are two lists: 'Voci legenda (serie)' and 'Etichette asse orizzontale (categoria)'. The 'Voci legenda (serie)' list has checkboxes for 'Olive', 'Salumi', 'formaggi', and 'Arance'. The 'Etichette asse orizzontale (categoria)' list has checkboxes for 'Italia', 'Francia', 'Grecia', and 'Spagna'. A red arrow points from the 'Arance' entry in the legend to the 'Modifica serie' dialog box.

	A	B	C	D	E
1		Italia	Francia	Grecia	Spagna
2	Olive	1000	200	700	800
3	Salumi	2000	500	100	1800
4	formaggi	3000	4300	870	350
5	Arance	500	200	170	600

The 'Modifica serie' dialog box shows the series name as '=Foglio5!\$A\$5' and the values as '=Foglio5!\$B\$5:\$E\$5'. The 'OK' button is highlighted.

- Nome serie: =Foglio5!\$A\$2
- Valori serie: =Foglio5!\$B\$2:\$E\$2

Ordinare i dati

- Menu > Ordina e filtra
 - Ordina per colonna, per riga, etc.

The screenshot shows the Microsoft Excel interface. The 'Ordina e filtra' option in the 'Celle' group of the ribbon is highlighted with a red box. A red arrow points from this box to the 'Ordina' dialog box. The dialog box is open, showing the 'Ordina per' dropdown menu with 'Data registrazione' selected. The 'Ordina in base a' dropdown is set to 'Valori', and the 'Ordine' dropdown is set to 'Dalla A alla Z'. The background shows a spreadsheet with columns A, B, and C, and rows 1 through 11. The data in the spreadsheet is as follows:

Data registrazione	Rifiuto	Peso (Kg)
ago-15	15 - Rifiuti di imballaggio, assorbenti, stracci, materiali filtranti e indumenti protettivi (non specificati altrimenti)	554,757.47
ago-15	16 - Rifiuti non specificati altrimenti nell'elenco	3,301.80
ago-15	17 - Rifiuti delle operazioni di costruzione e demolizione (compreso il terreno proveniente da siti contaminati)	68,354.60
ago-15	20 - Rifiuti urbani (rifiuti domestici e assimilabili prodotti da attività commerciali e industriali nonché dalle istituzioni) inclusi i rifiuti della	5,905,745.78
apr-15	15 - Rifiuti di imballaggio, assorbenti, stracci, materiali filtranti e indumenti protettivi (non specificati altrimenti)	465,260.26
apr-15	17 - Rifiuti delle operazioni di costruzione e demolizione (compreso il terreno proveniente da siti contaminati)	75,342.80
apr-15	20 - Rifiuti urbani (rifiuti domestici e assimilabili prodotti da attività commerciali e industriali nonché dalle istituzioni) inclusi i rifiuti della	5,415,146.11
feb-15	08 - Rifiuti della produzione, formulazione, fornitura ed uso di rivestimenti (pitture, vernici e smalti vetrat), adesivi, sigillanti e inchiostri	231.00
feb-15	15 - Rifiuti di imballaggio, assorbenti, stracci, materiali filtranti e indumenti protettivi (non specificati altrimenti)	401,009.24
feb-15	16 - Rifiuti non specificati altrimenti nell'elenco	739.20
	17 - Rifiuti delle operazioni di costruzione e demolizione (compreso il	

Filtrare i dati

- Menu > Ordina e filtra
 - Filtro per colonna, per riga, etc.

The screenshot shows the Microsoft Excel interface with the 'Ordina e filtra' menu open. The 'Filtro' option is selected, and the 'Filtro (CTRL+MAIUSC+L)' dialog box is displayed. The dialog box contains a list of columns with checkboxes next to them, indicating which columns are filtered. The 'Data registrazione' column is selected, and the 'Filtro' checkbox is checked. The dialog box also includes a search bar and a 'Cerca in (Tutto)' dropdown menu.

Data registrazione	Rifiuto	Peso (Kg)
ago-15	19 - rifiuti di imballaggio, assorbenti, stracci, materiali filtranti e	554,757.47
ago-15	1	3,301.80
ago-15	2	68,354.60
ago-15	2	5,905,745.78
apr-15	1	465,260.26
apr-15	2	75,342.80
apr-15	2	5,415,146.11
feb-15	1	231.00
feb-15	1	401,009.24
feb-15	1	739.20

Filtro (CTRL+MAIUSC+L)

Consente di attivare il filtro per le celle selezionate.

Dopo avere attivato il filtro, fare clic sulla freccia nell'instestazione della colonna per limitare i dati.

[Altre informazioni](#)

Funzioni – Massimo e Minimo (1)

=MAX(B2:B10)

- E' possibile trascinare la formula sulle altre serie (celle C11 e E11)

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1		Serie A	Serie B	Serie C	SerieD	
2		19	23	23	23	
3		8	54	24	23	
4		5	65	21	23	
5		234	43	5	23	
6		67	47	67	23	
7		45	36	69	23	
8		60	67	78	23	
9		1	78	72	23	
10		56	67	68	23	
11	Massimo	234				

Funzioni di conteggio (1)

- La funzione CONTA.NUMERI conteggia il numero di celle di un dato intervallo che contengono numeri (incluse le date)

The screenshot shows an Excel spreadsheet with the following data:

A	B	C	D	E
Data registrazione	Rifiuto	Peso (Kg)		
gen-15	08 - Rifiuti della produzione, formulazione, fornitura ed uso di rivestimenti (pitture, vernici e smalti vetrati), adesivi, sigillanti e inchiostri per stampa	333.90		
gen-15	15 - Rifiuti di imballaggio, assorbenti, stracci, materiali filtranti e indumenti protettivi (non specificati altrimenti)	415,783.71		
gen-15	16 - Rifiuti non specificati altrimenti nell'elenco	2,087.70	NUMERI(A2:C4)	
	17 - Rifiuti delle operazioni di costruzione e demolizione (compreso il			

The formula bar shows: `=CONTA.NUMERI(A2:C4)`

The zoomed-in view shows the following data:

C	D	E
Peso (Kg)		
inchiostri		
333.90		
415,783.71		
2,087.70		
eso il		
		3

Funzioni di conteggio (2)

- La funzione CONTA.VUOTE conteggia il numero di celle vuote di un dato intervallo
- E' possibile trascinare la formula sulle altre serie (celle C14 e E14)

	A	B	C	D	E	F
1		Serie A	Serie B	Serie C	SerieD	
2		19	23	23	23	
3		8	54	24	23	
4				21	23	
5		5	43	5	23	
6		234	47	67	23	
7		67	36	69		
8		45		78	23	
9		60	78	72	23	
10		1	67		23	
11		56	65	67		
12		4		7	23	
13			34	68	23	
14	Conta Vuoti	2	3	1	2	

Funzioni – Massimo e Minimo (2)

=MIN(B2:B10)

- E' possibile trascinare la formula sulle altre serie (celle C12 e E12)

	A	B	C	D	E	F
1		Serie A	Serie B	Serie C	SerieD	
2		19	23	23	23	
3		8	54	24	23	
4		5	65	21	23	
5		234	43	5	23	
6		67	47	67	23	
7		45	36	69	23	
8		60	67	78	23	
9		1	78	72	23	
10		56	67	68	23	
11	Massimo	234	78	78	23	
12	Minimo	1	23	5	23	
13						

Funzioni – Mediana, Moda (1)

- La MEDIANA è quel valore che divide in due parti uguali la distribuzione di una serie di dati
- La MODA è il valore che ha la frequenza più elevata in una distribuzione

Funzioni – Mediana, Moda (2)

La MEDIANA è quel valore che divide in due parti uguali la distribuzione di una serie di dati

	A	B	C	D	E	F
1		Serie A	Serie B	Serie D		
2		25	21	25		
3		30	25	30		
4		25	25	25		
5		40	25	40		
6		33	25	33		
7		25	29	25		
8		25	30	25		
9		29	33	29		
10		21	40	21		
11		42	42			
12	Mediana	27	27	25		

- Se ci sono due numeri in posizione centrale, restituisce la media dei due numeri (da C2 a C11 sono 10 valori, numero pari)

Funzioni – Mediana, Moda (3)

La MODA è il valore che ha la frequenza più elevata in una distribuzione

	A	B	C	D	E	F
1		Serie A	Serie B	Serie D		
2		25	21	25		
3		30	25	30		
4		25	25	25		
5		40	25	40		
6		33	25	33		
7		25	29	25		
8		25	30	25		
9		29	33	29		
10		21	40	21		
11		42	42			
12	Mediana	27	27	25		
13	Moda	25	25	25		

- E' il valore che compare più frequentemente in una serie numerica

Funzioni – Varianza e deviazione standard (1)

- Varianza e deviazione standard (o scarto quadratico medio) sono strumenti per analizzare gli indici di dispersione dei dati. Servono per valutare quanto una serie di valori sia più o meno uniforme rispetto alla sua media
- Le funzioni usate sono:
 - MEDIA
 - VAR.C
 - DEV.ST.C

Funzioni – Varianza e deviazione standard (2)

=MEDIA(B2:B10)

- E' possibile trascinare la formula sulle altre serie (celle C11 e D11)

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D
1		Serie A	Serie B	Serie C
2		19	23	23
3		8	54	24
4		5	65	21
5		234	43	5
6		67	47	67
7		45	36	69
8		60	67	78
9		1	78	72
10		56	67	68
11	Media	A(B2:B10)		

The 'Argomenti funzione' dialog box for the MEDIA function is open, showing the following details:

- Function: MEDIA
- Num1: B2:B10 (with a list of values: {19.8.5.234.67.45.60.1.56})
- Num2: (empty)
- Result: = 55
- Description: Restituisce la media aritmetica degli argomenti (numeri, nomi o riferimenti contenenti numeri).
- Num1 parameter description: Num1: num1;num2;... sono da 1 a 255 argomenti numerici di cui calcolare la media.
- Formula result: Risultato formula = 55
- Buttons: OK, Annulla
- Link: Guida relativa a questa funzione

Funzioni – Varianza e deviazione standard (3)

=DEV.ST.C(B2:B10)

- E' possibile trascinare la formula sulle altre serie (celle C12 e D12)

	A	B	C	F	G	H	I	J	K	L	M	N	O
1		Serie A	Serie B										
2		19											
3		8											
4		5											
5		234											
6		67											
7		45											
8		60											
9		1											
10		56											
11	Media	55	53,33333333	47,44444									
12	DevStandard	=DE											

NOTE:

- *DEV.ST.C* si applica per il calcolo su un campione
- *DEV.ST.P* si applica se si considera un'intera popolazione.

Funzioni – Varianza e deviazione standard (3)

=VAR.C(B2:B10)

- E' possibile trascinare la formula sulle altre serie (celle C13 e E13)

	A	B	C	D	E
1		Serie A	Serie B	Serie C	SerieD
2		19	23	23	23
3		8	54	24	23
4		5	65	21	23
5		234	43	5	23
6		67	47	67	23
7		45	36	69	23
8		60	67	78	23
9		1	78	72	23
10		56	67	68	23
11	Media	55	53,33333333	47,44444	23
12	DevStandard	71,7913644	17,62810256	28,40383	0
13	Varianza	5154	310,75	806,7778	0

NOTE:

- *VAR.C* si applica per il calcolo su un campione
- *VAR.P* si applica se si considera un'intera popolazione.

Funzioni matematiche e trigonometriche

- SIN, COS, COT, TAN, ...
- ABS, SOMMA, FATTORIALE, EXP, MCD, MCM, DISPARI, PARI, PRODOTTO, POTENZA, ...
- ARROTONDA.ECCESSO, ARROTONDA.DIFETTO, ...
- INT,
- MATR.DETERMINANTE, MATR.INVERSA, MATR.PRODOTTO, ...

Funzioni statistiche

- MEDIA, CONTA.VUOTE, CORRELAZIONE, CONTA.VALORI, DEV.ST.C, DEV.ST.P, VAR.C, VAR.P., ...
- MAX, MIN, MEDIANA, NORMALIZZATO, ...

Lavorare con le stringhe (1)

- CONCATENA(testo1;testo2;...)

DEV.ST ✕ ✓ fx =CONCATENA(A1;B1;C1;D1)

	A	B	C	D	E
1	Oggi	piove	a	dirotto	Oggipioveadirotto
2					Oggi piove a diretto
3					Oggi piove a diretto
4					

=CONCATENA(A1;" ";B1;" ";C1;" ";D1)

=CONCAT("Oggi";" ";"piove";" ";"a";" ";"dirotto")

Lavorare con le stringhe (2)

- IDENTICO(testo1;testo2)

Formula bar: E4 =IDENTICO(A1;A4)

	A	B	C	D	E
1	Oggi	piove	a	dirotto	Oggipioveadirotto
2					Oggi piove a dirotto
3					Oggi piove a dirotto
4	Oggi	Prendo la macchina			VERO
5					FALSO
6					

Formula bar: =IDENTICO(B1;A4)

Lavorare con le stringhe (3)

- STRINGA.ESTRAI(testo; inizio; num_caratteri)
- inizio = carattere da cui partire
- num_caratteri = numero di caratteri da estrarre dalla stringa
- Esempi:
 - STRINGA.ESTRAI(“Prendo la macchina”;1;6)
 - STRINGA.ESTRAI(B8;1;6)
 - STRINGA.ESTRAI(B8;8;2)

Excel formula bar: E8 : fx =STRINGA.ESTRAI(B8;1;6)

	A	B	C	D	E
7					
8	Oggi	Prendo la macchina			Prendo

Excel formula bar: E9 : fx =STRINGA.ESTRAI(B8;8;2)

	A	B	C	D	E
7					
8	Oggi	Prendo la macchina			Prendo
9					la
10					

Lavorare con le stringhe (4)

- TROVA(testo; stringa; inizio)
 - testo = testo da cercare
 - Stringa = stringa in cui cercare il testo
 - Inizio = numero del carattere della stringa da cui iniziare a cercare
- Consente di individuare una stringa di testo all'interno di una seconda stringa di testo e restituisce il numero corrispondente alla posizione iniziale della prima stringa di testo dal primo carattere della seconda stringa di testo

	A	B	C	D	E
7					
8	Oggi	Prendo la macchina			Prendo
9					la
10					11
11					

Lavorare con le stringhe (5)

- Composizione di funzioni
- `STRINGA.ESTRAI(testo; inizio; num_caratteri)` e `TROVA(testo; stringa; inizio)`
- ESEMPIO:
 - `STRINGA.ESTRAI(A11;1;TROVA("#";A11;1)-1)`
 - Estrae testo dalla posizione 1 alla posizione di "#" nella cella A2 (Isolanti in ceramica)

	A	B	C	D
11	Isolanti in ceramica #124-TD45-87			
12		Isolanti in ceramica		
13				

Lavorare con le stringhe (6)

- SINISTRA(testo; [num_caratt])
 - testo = testo di partenza
 - num_caratteri = numero di caratteri da estrarre
- Restituisce il primo carattere o i primi caratteri di una stringa di testo in base al numero di caratteri specificato a partire da sinistra

	A	B
14		
15	Esempio di stringa	Esempio di

Lavorare con le stringhe (7)

- DESTRA(testo; [num_caratt])
 - testo = testo di partenza
 - num_caratteri = numero di caratteri da estrarre
- Restituisce il primo carattere o i primi caratteri di una stringa di testo in base al numero di caratteri specificato a partire da destra

	A	B	C
14			
15	Esempio di stringa	Esempio di	
16	Modello_1	Modello	1

Lavorare con le stringhe (8)

- Copiare formule...

	A	B	C
14			
15	Esempio di stringa	Esempio di	
16	Modello_1	Modello	1
17	Modello_2	Modello	2
18	Modello_3	Modello	3
19	Modello_4	Modello	4
20	Modello_5	Modello	5
21	Modello_6	Modello	6

Formula bar: C16 \times \checkmark *fx* =DESTRA(A16;1)

Lavorare con le stringhe (9)

- LUNGHEZZA(testo)
 - Restituisce il numero di caratteri di una stringa di testo (spazi compresi)

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D
1	Stringa di testo	16		
2				
3				

The formula bar at the top shows the formula `=LUNGHEZZA(A1)` and the result `16`.

Lavorare con le stringhe (10)

- MINUSC(testo)
- Converte in minuscolo tutte le lettere maiuscole contenute in una stringa di testo

	A	B	C
1	Stringa di testo	16	
2	Nome Cognome	nome cognome	

- MAIUSC(testo)
- Converte in minuscolo tutte le lettere maiuscole contenute in una stringa di testo)

	A	B	C
1	Stringa di testo	16	
2	Nome Cognome	nome cognome	
3		NOME COGNOME	

Lavorare con le stringhe (11)

- RIMPIAZZA(testo_prec; inizio; num_caratt; nuovo_testo)
- Sostituisce parte di una stringa di testo con una stringa di testo diversa, in base al numero di caratteri specificati.

	A	B	C
4			
5	Scrivo una stringa di testo		
6	Scrivo una stringa di testo	Scrivo una nuova stringa di testo	
7	Scrivo una stringa di testo	Ciao! Scrivo una stringa di testo	
8	Scrivo una stringa di testo		

=RIMPIAZZA(A6;1;0; "Ciao! ")

Lavorare con le stringhe (12)

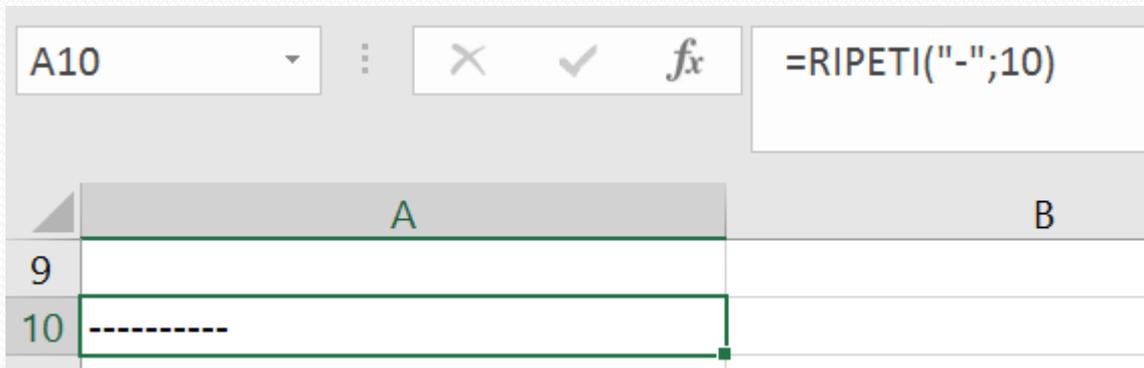
- SOSTITUISCI(testo; testo_prec; nuovo_testo; [occorrenza])
- Sostituisce nuovo_testo a testo_prec in una stringa di testo

	A	B
5	Scrivo una stringa di testo	
6	Scrivo una stringa di testo	Scrivo una nuova stringa di testo
7	Scrivo una stringa di testo	Ciao! Scrivo una stringa di testo
8	Scrivo una stringa di testo	Scrivo una ---- di testo

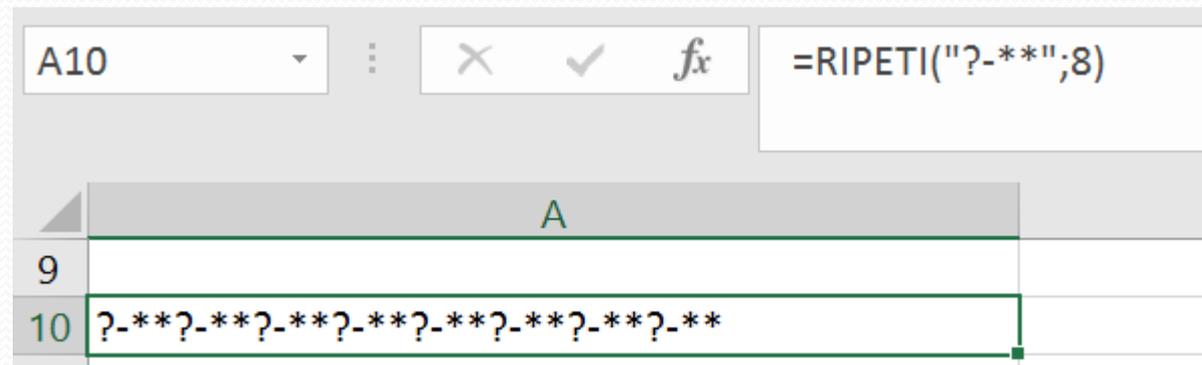
Formula bar: `=SOSTITUISCI(A8; "stringa";"----")`

Lavorare con le stringhe (13)

- RIPETI(testo; volte)
- Ripete un testo per il numero di volte specificato. Utilizzare la funzione RIPETI per riempire una cella con una stringa di testo ripetuta più volte.



The screenshot shows the Excel formula bar with the formula `=RIPETI("-",10)` entered. Below the formula bar, the spreadsheet grid shows column A and B. Row 9 is empty. Row 10, cell A10, contains ten hyphen characters (-----).



The screenshot shows the Excel formula bar with the formula `=RIPETI("?-*");8)` entered. Below the formula bar, the spreadsheet grid shows column A. Row 9 is empty. Row 10, cell A10, contains eight repetitions of the string "?-*" concatenated together: "?-*?-*?-*?-*?-*?-*?-*?-*".

Lavorare con le stringhe (14)

- TESTO(Valore da formattare;"Codice formato da applicare")

	A	B
11		
12	20/05/17	
13	20/05/17 18.40	
14	28,5%	

=TESTO(0,285;"0,0%")

Lavorare con le stringhe (15)

- TESTO.UNISCI(delimitatore;ignora_vuote; testo1;[testo2]; ...)
- La funzione TESTO.UNISCI combina il testo di più intervalli e/o stringhe e include un delimitatore specificato dall'utente tra ogni valore di testo da unire. Se il delimitatore è una stringa di testo vuota, la funzione concatena correttamente gli intervalli

	A	B	C
15			
16	Descrizione del prodotto	Modello	Unione
17	Giacca Uomo	#go31A23	Giacca Uomo #go31A23
18	Pantalone lungo donna	#pld5378	Pantalone lungo donna #pld5378
19	Pantalone corto uomo	#pcu3425	Pantalone corto uomo #pcu3425
20	Abito donna	#ad9496	Abito donna #ad9496
21	Pantalone bambino	#pb04531	Pantalone bambino #pb04531

Lavorare con le stringhe (16)

- ANNULLA.SPAZI(testo)
- Rimuove tutti gli spazi dal testo ad eccezione dei singoli spazi tra le parole. Utilizzare la funzione ANNULLA.SPAZI sul testo creato con altre applicazioni che può presentare una distribuzione irregolare degli spazi

	A	B
23		
24		
25	Testo con spazi 0 9	Testo con spazi 0 9

Funzioni Logiche (1)

- TRUE
- FALSE
- SE (IF)
 - SE(qualcosa è Vero; fai qualcosa; altrimenti fai qualcos'altro)

	A	B	C	D	E	F
15					Costo ma per un capo:	90
16	Descrizione del prodotto	Modello	Unione		Costo	
17	Giacca Uomo	#go31A23	Giacca Uomo #go31A23		100,00	NO
18	Pantalone lungo donna	#pld5378	Pantalone lungo donna #pld5378		80,00	compra
19	Pantalone corto uomo	#pcu3425	Pantalone corto uomo #pcu3425		60,00	compra
20	Abito donna	#ad9496	Abito donna #ad9496		110,00	NO
21	Pantalone bambino	#pb04531	Pantalone bambino #pb04531		40,00	compra

Funzioni Logiche (2)

- NON(Argomento)
- Inverte la logica dell'argomento (Restituisce FALSO per un argomento VERO e VERO per un argomento FALSO)

	A	B	C	D	E	F	G
15					Costo ma per un capo:	90	
16	Descrizione del prodotto	Modello	Unione		Costo		
17	Giacca Uomo	#go31A23	Giacca Uomo #go31A23		100,00	NO	VERO
18	Pantalone lungo donna	#pld5378	Pantalone lungo donna #pld5378		80,00	compra	VERO
19	Pantalone corto uomo	#pcu3425	Pantalone corto uomo #pcu3425		60,00	compra	FALSO
20	Abito donna	#ad9496	Abito donna #ad9496		110,00	NO	VERO
21	Pantalone bambino	#pb04531	Pantalone bambino #pb04531		40,00	compra	VERO

Funzioni Logiche (2)

- NON(Argomento)
- Inverte la logica dell'argomento (Restituisce FALSO per un argomento VERO e VERO per un argomento FALSO)

	A	B	C	D	E	F	G
15					Costo ma per un capo:	90	
16	Descrizione del prodotto	Modello	Unione		Costo		
17	Giacca Uomo	#go31A23	Giacca Uomo #go31A23		100,00	NO	VERO
18	Pantalone lungo donna	#pld5378	Pantalone lungo donna #pld5378		80,00	compra	VERO
19	Pantalone corto uomo	#pcu3425	Pantalone corto uomo #pcu3425		60,00	compra	FALSO
20	Abito donna	#ad9496	Abito donna #ad9496		110,00	NO	VERO
21	Pantalone bambino	#pb04531	Pantalone bambino #pb04531		40,00	compra	VERO

Funzioni Logiche (3)

- $O(\text{arg1};\text{arg1};\dots)$
- Restituisce VERO se un argomento qualsiasi è VERO

	A	B	C
27			
28	VALORI		
29	50		VERO
30	100		FALSO

Formula bar: $=O(A29<90;A29>1)$

	A	B	C
27			
28	VALORI		
29	50		VERO
30	100		FALSO

Formula bar: $=O(A30<90)$

Funzioni Logiche (4)

- =SWITCH(Valore da cambiare, Valore per corrispondenza1...[2-126], Valore da restituire se esiste una corrispondenza1...[2-126], Valore da restituire se non esistono corrispondenze)
- La funzione SWITCH valuta un valore, chiamato **espressione**, rispetto a un elenco di valori e restituisce il risultato che equivale al primo valore corrispondente. Se non ci sono valori corrispondenti, verrà restituito un valore predefinito facoltativo

B33 :    =SWITCH(A33;1;"Domenica";2;"Lunedì";3;"Martedì";"Nessuna corrispondenza")

	A	B	C	D	E	F	G	H
31								
32	VALORE	Risultato		VALORE				
33		1 Domenica		20/05/2017	Domenica			
34		2 Lunedì		21/05/2017	Lunedì			
35		3 Martedì		22/05/2017	Martedì			
36		4 Nessuna corrispondenza		23/05/2017	Mercoledì			
37		5 Nessuna corrispondenza		24/05/2017	Giovedì			
38		6 Nessuna corrispondenza		25/05/2017	Venerdì			
39		7 Nessuna corrispondenza		26/05/2017	Sabato			

=SWITCH(GIORNO.SETTIMANA(A34);1;"Domenica";2;"Lunedì";3;"Martedì";4;"Mercoledì";5;"Giovedì";6;"Venerdì";7;"Sabato")

Lavorare con i numeri (1)

- FISSO(num; [decimali]; [nessun_separatore])
 - Arrotonda un numero al numero specificato di decimali, formattandolo con i separatori delle migliaia e la virgola decimale, e restituisce il risultato in forma di testo.

	A	B	C
13	123,6756757	123,676	Arrotonda il numero in A4 con due cifre a sinistra della virgola decimale.
14			

Mappe 3D (1)

- Selezionare dati > Inserisci > Mappa 3D (componenti aggiuntivi)

The screenshot shows the Microsoft Excel interface with the 'Mappe 3D' ribbon selected. The ribbon includes options like 'Linee', 'Istogramma', 'Positivi/negativi', 'Filtri', 'Collegamento ipertestuale', 'Testo', and 'Simboli'. A dropdown menu for 'Mappa 3D' is open, showing 'Apri Mappe 3D' and 'Aggiungi i dati selezionati a Mappe 3D'. A tooltip for 'Apri Mappe 3D' is displayed, explaining that it opens 3D tours and allows creating new ones, provided the data contains geographic information like addresses or countries. A link for 'Altre informazioni' is also present.

Città	Popolazione
Firenze	361.000
Milano	1.251.000
Roma	2.627.000
Napoli	960.069
Bologna	375.893

Apri Mappe 3D

Apri i tour di Mappe 3D o consente di crearne di nuovi. Assicurarsi che la cartella di lavoro contenga dati geografici, come indirizzi o paesi.

[? Altre informazioni](#)

Mappe 3D (2)



Tour 1

Scena 1 (10 sec)

bing

© 2017 HERE

Elenco campi

Trascinare i campi nel Riquadro livello.

- Intervallo
 - Città
 - Popolazione

Aggiungi livello

Livello 1

Dati

Posizione 100%

Città Provincia

+ Aggiungi campo

Altezza

Popolazione (Somma)

+ Aggiungi campo

Categoria

+ Aggiungi campo

Tempo

+ Aggiungi campo

Mappe 3D (1)



▼ Popolazione

▼ Dati

Posizione 80%

Città Provincia

+ Aggiungi campo

Valore

Popolazione (Somma)

+ Aggiungi campo

Categoria

+ Aggiungi campo

Tempo

+ Aggiungi campo

Macro (1)

- Se si eseguono ripetutamente le stesse attività in Microsoft Excel, è possibile automatizzarle registrando una macro.
- Una macro è una azione o un insieme di azioni che è possibile eseguire per un numero illimitato di volte
- Quando si crea una macro, vengono registrati i clic del mouse e le sequenze di tasti
- Dopo aver creato una macro, è possibile modificarla per cambiarne lievemente il funzionamento.

Macro (2)

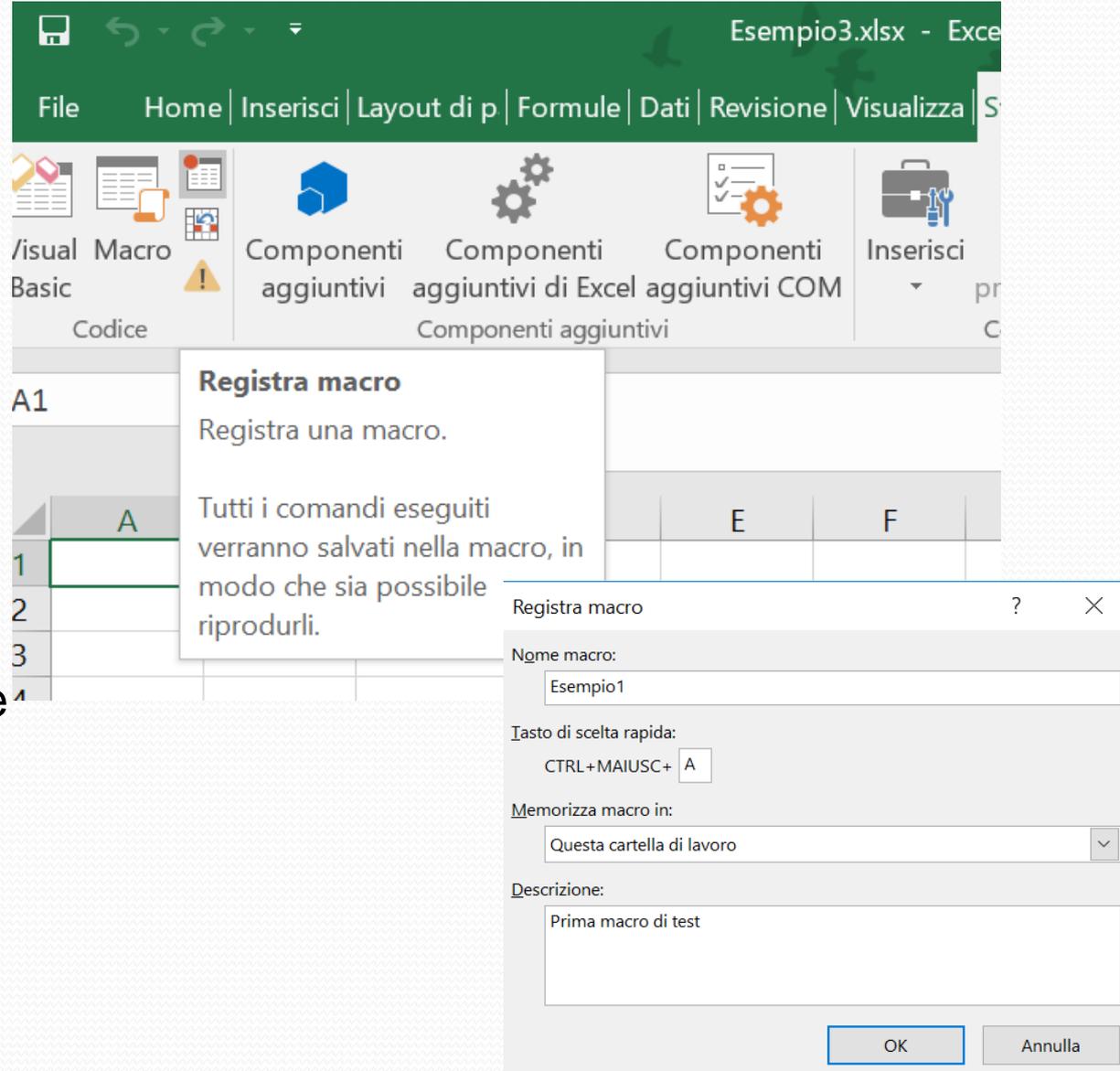
- Se si deve creare ogni mese un rapporto per il responsabile della contabilità in modo da:
 - formattare in rosso i nomi dei clienti con i conti scaduti e applicare inoltre il grassetto
 - Allora è possibile creare ed eseguire una macro per applicare rapidamente queste modifiche di formattazione alle celle selezionate

Macro (3)

- Attivare la scheda per lo sviluppo delle macro
- Verificare che nella barra multifunzione sia visualizzata la scheda **Sviluppo**.
- Se la barra **Sviluppo** non è visibile, eseguire le operazioni seguenti:
 - **File > Opzioni > Personalizzazione barra multifunzione** e Nell'elenco **Schede principali** della categoria **Personalizza barra multifunzione** fare clic su **Sviluppo** e quindi su **OK**

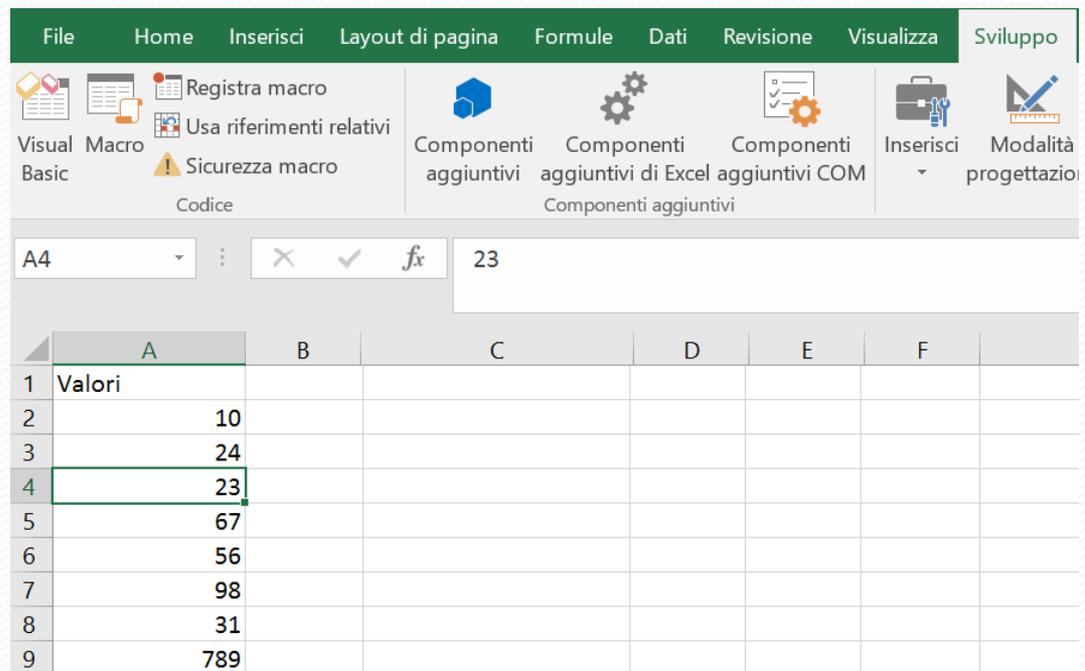
Macro (4)

- Registrare una macro
- Nel gruppo **Codice** della scheda **Sviluppo** fare clic su **Registra macro**
- Immettere:
 - **Nome macro**
 - **Tasto di scelta rapida**
 - **Descrizione**
- Fare clic su **OK** per avviare la registrazione
- Eseguire le operazioni
- Cliccare su interrompi registrazione
- Ri usare la macro quando necessario



Macro – Esempio soglia (1)

- Partenza:
 - Presenza di lista di valori interi in un foglio excel (una colonna, esempio: A2:A20)
- Scopo:
 - Creare una macro capace di determinare quali di questi valori superano una determinata soglia

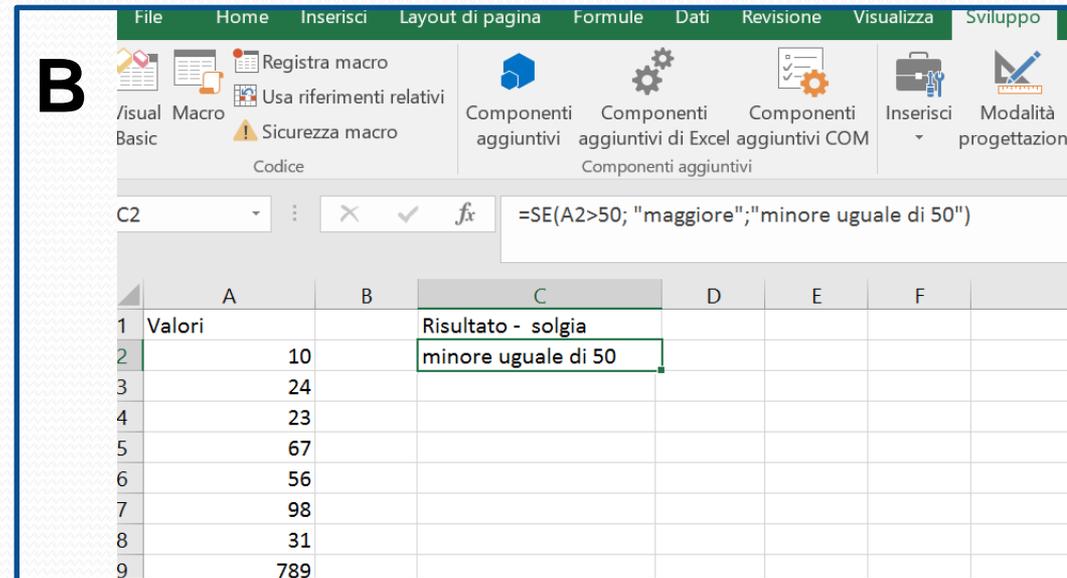
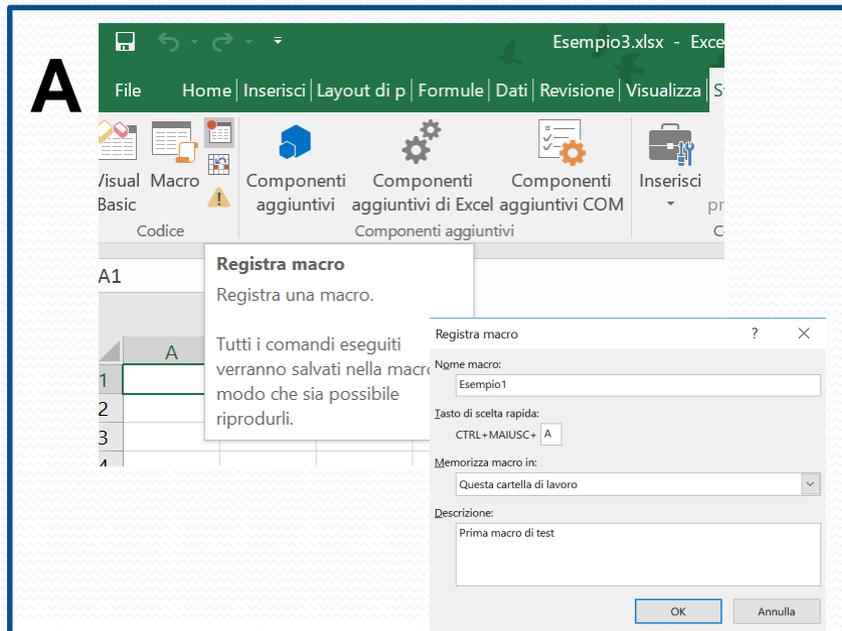


The screenshot shows the Microsoft Excel ribbon with the 'Sviluppo' (Developer) tab selected. The 'Macro' group is active, showing options like 'Registra macro', 'Usa riferimenti relativi', and 'Sicurezza macro'. Below the ribbon, the formula bar shows 'A4' and the value '23'. The spreadsheet below has a single column of data in column A, with the value 23 highlighted in row 4.

	A	B	C	D	E	F
1	Valori					
2	10					
3	24					
4	23					
5	67					
6	56					
7	98					
8	31					
9	789					

Macro – Esempio soglia (2)

- Svolgimento:
 - A. Registrare una macro (nome, etc.)
 - B. Scrivere nella cella B2 la formula:
`=SE(A2>50; "maggiore";"minore uguale di 50")`
 - C. Trascinare la formula in tutta la colonna B (B2:B20)
 - D. Interrompere la registrazione



Macro – Esempio copia/incolla (3)

- Svolgimento:
 - A. Registrare una macro (nome, etc.)
 - B. Scrivere nella cella B2 la formula:
`=SE(A2>50; "maggiore"; "minore uguale di 50")`
 - C. Trascinare la formula in tutta la colonna B (B2:B20)
 - D. Interrompere la registrazione

C

	A	B	C	D	E	F
1	Valori		Risultato - solgia			
2	10		minore uguale di 50			
3	24		minore uguale di 50			
4	23		minore uguale di 50			
5	67		maggiore			
6	56		maggiore			
7	98		maggiore			
8	31		minore uguale di 50			
9	789		maggiore			

D

Interrompi registrazione
Registra una macro.
Tutti i comandi eseguiti verranno salvati nella macro, in modo che sia possibile riprodurli.

	A	B	C	D	E	F
1	Valori		solgia			
2			minore uguale di 50			
3			minore uguale di 50			
4			minore uguale di 50			
5			maggiore			
6			maggiore			
7			maggiore			
8			minore uguale di 50			

Macro – Esempio soglia (4)

- Uso:
 - Andare in un nuovo foglio excel
 - Partire da una nuova lista di valori
 - Eseguire la macro:
Sviluppo >
Macro >
scegliere la macro

Macro dialog box showing the list of macros. The macro 'soglia' is selected. The 'Esegui' button is highlighted with a red arrow.

Excel spreadsheet showing the formula bar and the resulting data. The formula bar contains the formula: `=SE(A2>50;"Maggiore";"Minore o uguale")`. The resulting data is shown in column B.

A	B	C	D
Valori NUOVI	Risultato - Soglia		
	34 Minore o uguale		
	20 Minore o uguale		
	23 Minore o uguale		
	67 Maggiore		
	5 Minore o uguale		
	98 Maggiore		
	31 Minore o uguale		
	9 Minore o uguale		
	23 Minore o uguale		
	45 Minore o uguale		
	68 Maggiore		

Macro – Esempio soglia (5)

Visual Basic Editor (VBE) interface showing the 'Macro' dialog box. The 'Nome macro:' field contains 'soglia'. The 'Esegui' button is highlighted. The background shows the Excel spreadsheet with the following data:

Valori NUOVI	Risultato - Soglia
34	Minore o uguale
20	Minore o uguale
23	Minore o uguale
67	Maggiore
5	Minore o uguale
98	Maggiore
31	Minore o uguale

- Modifica di una macro tramite editor Visual Basic

Visual Basic Editor (VBE) interface showing the code for the 'soglia' macro. The code is as follows:

```
Application.CutCopyMode = False
ActiveCell.FormulaR1C1 = "=Foglio5!RC[-1]=IF(RC[-2]>50, ""maggiore"", ""minore uguale di 50"")"
Range("C2").Select
ActiveCell.FormulaR1C1 = "=IF(RC[-2]>50, ""maggiore"", ""minore uguale di 50"")"
Range("C3").Select
ActiveCell.FormulaR1C1 = "=IF(RC[-2]>50, ""maggiore"", ""minore uguale di 50"")"
Range("C1").Select
ActiveCell.FormulaR1C1 = "Risultato - soglia"
Range("C2").Select
End Sub
Sub soglia()
' soglia Macro
' Scelta rapida da tastiera: CTRL+MAIUSC+B
ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50, ""Maggiore"", ""Minore o uguale"")"
Range("B2").Select
ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50, ""Maggiore"", ""Minore o uguale"")"
Range("B2").Select
Selection.AutoFill Destination:=Range("B2:B20"), Type:=xlFillDefault
Range("B2:B20").Select
End Sub
Sub Macro5()
' Macro5 Macro
' Scelta rapida da tastiera: CTRL+MAIUSC+D
End Sub
```

Macro – Esempio soglia (6)

- Variante:
 - E' possibile anche registrare le operazioni di formattazione alle celle.
 - Ad esempio 'mettere sfondo giallo alle caselle che fanno riferimento a valori minori o uguali a 50'

	A	B	C	D	E	F
1	Valori	Soglia				
2		67 Maggiore				
3		675 Maggiore				
4		34 Minore o uguale				
5		34 Minore o uguale				
6		13 Minore o uguale				
7		78 Maggiore				
8		95 Maggiore				
9		44 Minore o uguale				
10		56 Maggiore				
11		86 Maggiore				

Macro – Esempio soglia (7)

- Richiamare la Macro agganciandola ad un bottone

The screenshot shows the Microsoft Excel interface with the 'Sviluppo' (Developer) ribbon active. The 'Controlli modulo' (Form Controls) dropdown menu is open, showing various control options. The spreadsheet data is as follows:

	A	B	C	D	E	F	G
1	Valori						
2	67						
3	675						
4	34						
5	34						
6	13						
7	78						

Macro – Esempio soglia (8)

Assegna macro

Nome macro:

- Soglia_visualizzazione
- copialncolla
- Esempio1
- Macro5
- soglia
- Soglia_visualizzazione

Modifica

Registra...

	A	B	C	D	E	F
1	Valori	Soglia		Pulsante		
2	67					
3	675					
4	34					
5	34					
6	13					
7	78					
8	95					

Premi per la soglia

- Taglia
- Copia
- Incolla
- Modifica testo
- Gruppo
- Ordine
- Assegna macro...
- Formato controllo...

- Associazione bottone-Macro
- Modifica vista bottone
- Click sul bottone per eseguire macro

Visual Macro Basic

Registra macro

Usa riferimenti relativi

Sicurezza macro

Codice

Componenti aggiuntivi

Componenti aggiuntivi di Excel

Componenti aggiuntivi COM

Inserisci

Modalità progettazione

H15

	A	B	C	D	E	F	G	H
1	Valori							
2	10	Minore o uguale						
3	24	Minore o uguale			Premi per la soglia			
4	23	Minore o uguale						
5	67	Maggiore						
6	56	Maggiore						
7	98	Maggiore						
8	31	Minore o uguale						
9	789	Maggiore						
0	23	Minore o uguale						
1	45	Minore o uguale						
2	68	Maggiore						
3	87	Maggiore						
4	45	Minore o uguale						
5	72	Maggiore						
6	21	Minore o uguale						
7	98	Maggiore						
8	67	Maggiore						
9	89	Maggiore						
0	802	Maggiore						

Tipi di pulsante

Tipo di controllo	Descrizione
 Etichetta	Testo aggiunto a un foglio di lavoro o a un modulo per fornire informazioni su un controllo, sul foglio di lavoro o sul modulo.
 Casella di gruppo	Bordo ed etichetta che raggruppano controlli correlati, quali pulsanti di opzione o caselle di controllo.
 Pulsante	Pulsante che esegue una macro quando viene scelto.
 Casella di controllo	Casella che attiva o disattiva un'opzione. In un foglio o in un gruppo è possibile selezionare contemporaneamente più caselle di controllo.
 Pulsante di opzione	Pulsante che seleziona un'opzione in un gruppo contenuto in una casella di gruppo. Poiché in un gruppo è possibile selezionare un solo pulsante di opzione, questo viene utilizzato quando è consentita una sola possibilità di scelta.
 Casella di riepilogo	Casella contenente un elenco di voci.
 Casella combinata	Casella di riepilogo a discesa. La voce selezionata nella casella di riepilogo è visualizzata nella casella di testo.
 Barra di scorrimento	Controllo che scorre un intervallo di valori quando si fa clic sulle frecce di scorrimento o quando si trascina la casella di scorrimento. È possibile muoversi in una successiva pagina di valori facendo clic tra la casella di scorrimento e la freccia di scorrimento.
 Casella di selezione	Pulsante con una freccia Su e una freccia Giù che è possibile associare a una cella e che consente di aumentare o diminuire un valore facendo clic sulla freccia corrispondente.



Gestire le macro



Gestire le macro

- Con l'Editor di Visual Basic è possibile:
 - modificare le macro
 - copiare macro da un modulo all'altro
 - copiare macro tra diverse cartelle di lavoro
 - rinominare i moduli che memorizzano le macro
 - rinominare le macro stesse
- Se ad esempio si desidera che la macro per la sottolineatura di alcune celle, sarà possibile registrare un'altra macro per formattare la cella, quindi copiare le istruzioni da quest'ultima nella prima macro

Modificare una macro (1)

- Supponiamo di partire dal codice per determinare quali valori superano una data soglia...

The screenshot displays the Microsoft Visual Basic Applications Edition window for 'Esempio4.xlsx'. The macro editor shows the following code:

```
Sub soglia()  
    glia Macro  
  
    Range("B2").Select  
    ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50, ""Maggiore"", ""Minore o uguale"")"  
    Range("B2").Select  
    Selection.AutoFill Destination:=Range("B2:B18"), Type:=xlFillDefault  
    Range("B2:B18").Select  
    Range("C2").Select  
End Sub
```

The background Excel spreadsheet shows the following data:

A	B
Valori	
56	Maggiore
657	Maggiore
76	Maggiore
33	Minore o uguale
234	Maggiore
43	Minore o uguale
32	Minore o uguale
23	Minore o uguale
678	Maggiore
56	Maggiore
9	Minore o uguale
77	Maggiore
80	Maggiore
34	Minore o uguale
76	Maggiore
12	Minore o uguale
1	Minore o uguale

Modificare una macro (2)

- Modifica del titolo di una macro
- Commenti (con apice singolo)

The screenshot shows the Microsoft Visual Basic for Applications editor window. The title bar reads "Microsoft Visual Basic, Applications Edition - Esempio4.xlsx - [Modulo1 (codice)]". The menu bar includes "File", "Modifica", "Visualizza", "Inserisci", "Formato", "Debug", "Esegui", "Strumenti", "Aggiunte", and "Finestra". The status bar indicates "Riga 14, Col 8".

The left sidebar shows the "Progetto - VBAProject" tree with "VBAProject (Esempio4.xlsx)" expanded to show "Modulo1". The "Proprietà - Modulo1" pane is also visible.

The main editor window displays the code for the macro "NuovoTitoloMacro" in the "generale" view. The code is as follows:

```
Sub NuovoTitoloMacro()  
'  
' commenti con apice singolo....  
' Nuovo titolo della Marco: "NuovoTitoloMacro"  
' Nuovo titolo della Marco: "soglia"  
'  
Range("B2").Select  
ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50, ""Maggiore"", ""Minore o uguale""  
Range("B2").Select  
Selection.AutoFill Destination:=Range("B2:B18"), Type:=xlFillDefault  
Range("B2:B18").Select  
' Range("C2").Select commento le righe di codice inutili  
End Sub
```

Two red boxes highlight specific parts of the code: the first box encloses the opening "Sub" line and the first three comment lines; the second box encloses the line "' Range("C2").Select commento le righe di codice inutili".

At the bottom, the "Espressioni di controllo" (Control Expressions) pane is visible, showing a table with columns for "Espressione", "Valore", "Tipo", and "Contesto".

Modificare una macro (3)

- Variazione del Range (celle su cui eseguire le operazioni)

```
(generale) NuovoTitoloMacro
Sub NuovoTitoloMacro()
'
' commenti con apice singolo....
' Nuovo titolo della Marco: "NuovoTitoloMacro"
' Nuovo titolo della Marco: "soglia"
'
Range("B2").Select
ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50,""Maggiore",""Minore o uguale""
' --variazione range da B2:B19 a B2:B18
Selection.AutoFill Destination:=Range("B2:B16"), Type:=xlFillDefault
Range("B2:B16").Select
' Range("C2").Select commento le righe di codice inutili
End Sub
```

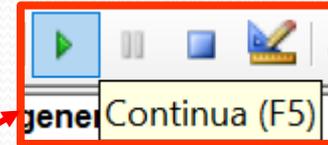


	A	B	C
1	Valori		
2	56	Maggiore	
3	657	Maggiore	
4	76	Maggiore	
5	33	Minore o uguale	
6	234	Maggiore	
7	43	Minore o uguale	
8	32	Minore o uguale	
9	23	Minore o uguale	
10	678	Maggiore	
11	56	Maggiore	
12	9	Minore o uguale	
13	77	Maggiore	
14	80	Maggiore	
15	34	Minore o uguale	
16	76	Maggiore	
17	12		
18	1		

- Sia il calcolo della 'soglia' che la selezione finale delle celle sono NON più nel range B2:B18 ma coinvolgono le celle B2:B16

Uso del debugger (1)

- E' possibile usare il debugger
- Uso dei breakpoint



Microsoft Visual Basic, Applications Edition - Esempio4.xlsx [interruzione] - [Modulo1 (codice)]

File Modifica Visualizza Inserisci Formato Debug Esegui Strumenti Aggiunte Finestra ?

Progetto - VBAProject

Microsoft Excel Oggetti

- VBAProject (Esempio4.xlsx)
 - Foglio1 (Sheet1)
 - Foglio2 (Foglio1)
 - Questa cartella di lavoro
 - Moduli
 - Modulo1

Proprietà - Modulo1

Modulo1 Modulo

Alfabetico Per categoria

(Name) Modulo1

(generale) NuovoTitoloMacro

```
Sub NuovoTitoloMacro ()
' commenti con apice singolo...
' Nuovo titolo della Marco: "NuovoTitoloMacro"
' Nuovo titolo della Marco: "soglia"
'
'
' Range("B2").Select
ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50, ""Maggiore"", ""Minore o uguale""
' --variazione range da B2:B19 a B2:B18
' Selection.AutoFill Destination:=Range("B2:B16"), Type:=xlFillDefault
Range("B2:B16").Select
' Range("C2").Select commento le righe di codice inutili
End Sub
```

Espressioni di controllo

Espressione	Valore	Tipo	Contesto
-------------	--------	------	----------

Uso del debugger (2)

- Si preme due volte su F5:
 - il debugger si ferma al secondo breakpoint
 - Il debugger ha eseguito la prima riga di codice sul foglio excel (ovvero ha calcolato la soglia nella cella B2)
 - Il debugger deve ancora eseguire le altre istruzioni

The screenshot displays the VBA editor for 'Esempio4.xlsx'. The background Excel spreadsheet shows a column of values in column A (1 to 18) and a column of results in column B (1 to 18). The VBA editor shows a macro named 'NuovoTitoloMacro' with the following code:

```
Sub NuovoTitoloMacro()  
    ' commenti con apice singolo....  
    ' Nuovo titolo della Marco: "NuovoTitoloMacro"  
    ' Nuovo titolo della Marco: "soglia"  
    '   
    Range("B2").Select  
    ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50, ""Maggiore"", ""Minore o uguale""  
    '--variazione range da B2:B19 a B2:B18  
    Selection.AutoFill Destination:=Range("B2:B16"), Type:=xlFillDefault  
    Range("B2:B16").Select  
    Range("C2").Select commento le righe di codice inutili  
End Sub
```

The debugger is currently stopped at the line 'Selection.AutoFill Destination:=Range("B2:B16"), Type:=xlFillDefault'. The 'Proprietà - Modulo1' window shows the 'Modulo1' module.

Uso del debugger (3)

- Fine della esecuzione:
 - Tutte le operazioni VB sono state eseguite

The screenshot displays the Microsoft Visual Basic Applications Edition interface. On the left, an Excel spreadsheet is visible with a column of values: 56 Maggiore, 657 Maggiore, 76 Maggiore, 33 Minore o uguale, 234 Maggiore, 43 Minore o uguale, 32 Minore o uguale, 23 Minore o uguale, 678 Maggiore, 56 Maggiore, 9 Minore o uguale, 77 Maggiore, 80 Maggiore, 34 Minore o uguale, 76 Maggiore, 12, and 1. The VBA editor window shows the 'NuovoTitoloMacro' subroutine with the following code:

```
Sub NuovoTitoloMacro()  
    ' commenti con apice singolo...  
    ' Nuovo titolo della Marco: "NuovoTitoloMacro"  
    ' Nuovo titolo della Marco: "soglia"  
    ' Range("B2").Select  
    ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50,"Maggiore","Minore o uguale")  
    ' --variazione range da B2:B19 a B2:B18  
    Selection.AutoFill Destination:=Range("B2:B16"), Type:=xlFillDefault  
    Range("B2:B16").Select  
    ' Range("C2").Select commento le righe di codice inutili  
End Sub
```

The 'Espressioni di controllo' window at the bottom is empty, indicating the macro has finished execution.

Modificare una macro (4)

- Partendo da un esempio simile e quindi sempre da una lista di valori nelle celle A2:A18
- Si inizia la registrazione della macro
- Si scrive il titolo della colonna B (cella B1)
- Si cambia lo sfondo della cella B1
- Si selezionano le celle dei valori (A2:A18) e si ordinano per valore (dal più piccolo al più grande da menu' > dati>ordina)
- Si scrive la formula per la soglia nella cella B2
- Si trascina la formula nelle altre celle (B3:B18)
- Si finisce la registrazione della macro

Modificare una macro (5)

- Applicazione della macro: stato iniziale

	A	B	C	D	E	F	G	H	I
1	Valori								
2	56								
3	657								
4	76								
5	33								
6	234								
7	43								
8	32								
9	23								
10	678								
11	56								
12	9								
13	77								
14	80								
15	34								
16	76								
17	12								
18	1								
19									
20									
21									

Macro

Nome macro:

SogliaSottolineaturaOrdinamento

NuovoTitoloMacro

SogliaSottolineaturaOrdinamento

Macro in: Tutte le cartelle di lavoro aperte

Descrizione

Esegui

Esegui istruzione

Modifica

Crea

Elimina

Opzioni...

Annulla

Modificare una macro (6)

- Applicazione della macro: stato finale

B2 X ✓ fx =SE(A2>50;"Maggiore";"Minore o uguale")

	A	B	C	D	E	F	G	H	I
1	Valori	Maggiore di 50							
2	1	Minore o uguale							
3	9	Minore o uguale							
4	12	Minore o uguale							
5	23	Minore o uguale							
6	32	Minore o uguale							
7	33	Minore o uguale							
8	34	Minore o uguale							
9	43	Minore o uguale							
10	56	Maggiore							
11	56	Maggiore							
12	76	Maggiore							
13	76	Maggiore							
14	77	Maggiore							
15	80	Maggiore							
16	234	Maggiore							
17	657	Maggiore							
18	678	Maggiore							
19									
20									
21									

Macro

Nome macro:

- SogliaSottolineaturaOrdinamento
- NuovoTitoloMacro
- SogliaSottolineaturaOrdinamento

Macro in: Tutte le cartelle di lavoro aperte

Descrizione

Esegui

Esegui istruzione

Modifica

Crea

Elimina

Opzioni...

Annulla

Modificare una macro (7)

- Si scrive il titolo in B1
- Si Allarga la colonna B (Section.ColumnWidth)

The screenshot displays the Microsoft Excel interface and the Visual Basic Editor (VBE) for 'Esempio4.xlsx'. In the Excel window, the 'Macro' ribbon is active, and the 'Codice' (Code) window is open. The spreadsheet shows column B with a title 'Maggiore di 50' in cell B1. The VBE window shows the 'SogliaSottolineatura' macro code. The code is as follows:

```
' Nuovo titolo della Marco: "NuovoTitoloMacro"  
' Nuovo titolo della Marco: "soglia"  
'  
Range("B2").Select  
ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50, ""Mac  
--variazione range da B2:B19 a B2:B18  
Selection.AutoFill Destination:=Range("B2:B16")  
Range("B2:B16").Select  
' Range("C2").Select commento le righe di codice  
End Sub  
Sub SogliaSottolineaturaOrdinamento()  
'  
' SogliaSottolineaturaOrdinamento Macro  
'  
' Scelta rapida da tastiera: CTRL+MAIUSC+C  
'  
Range("B1").Select  
Application.CutCopyMode = False  
ActiveCell.FormulaR1C1 = "Maggiore di 50"  
Columns("B:B").Select  
Selection.ColumnWidth = 17.27  
Range("B1").Select  
With Selection.Interior  
.Pattern = xlSolid  
.PatternColorIndex = xlAutomatic  
.Color = 65535  
.TintAndShade = 0  
.PatternTintAndShade = 0  
End With
```

The VBE Properties window for 'Modulo1' is also visible, showing the 'Name' property set to 'Modulo1'.

Modificare una macro (8)

- Uso del debugger per vedere le operazioni passo-passo

- Sfondo Cella B1

Microsoft Visual Basic, Applications Edition - Esempio4.xlsx [interruzione] - [Modulo1 (codice)]

File Modifica Visualizza Inserisci Formato Debug Esegui Strumenti Aggiunte Finestra ?

Progetto - VBAProject

VBAProject (Esempio4.xlsx)

- Microsoft Excel Oggetti
 - Foglio1 (Sheet1)
 - Foglio2 (Foglio1)
 - Foglio3 (Foglio2)
 - Foglio4 (Foglio3)
 - Questa_cartella_di_lavoro
- Moduli
 - Modulo1

Proprietà - Modulo1

Modulo1 Modulo

Alfabetico Per categoria

(Name) Modulo1

```

Range("B2").Select
ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50
--variazione range da B2:B19 a B2:B18
Selection.AutoFill Destination:=Range("
Range("B2:B16").Select
Range("C2").Select commento le righe di
End Sub
Sub SogliaSottolineaturaOrdinamento()
SogliaSottolineaturaOrdinamento Macro
Scelta rapida da tastiera: CTRL+MAIUSC+C
Range("B1").Select
Application.CutCopyMode = False
ActiveCell.FormulaR1C1 = "Maggiore di 5
Columns("B:B").Select
Selection.ColumnWidth = 17.27
Range("B1").Select
With Selection.Interior
.Pattern = xlSolid
.PatternColorIndex = xlAutomatic
.Color = 65535
.TintAndShade = 0
.PatternTintAndShade = 0
End With
Range("A2:A18").Select
ActiveWorkbook.Worksheets("Foglio2").Sc
    
```

FFFFFF	000000	333333	666666	999999	cccccc	CCCC99	9999CC	666699
669000	663300	996633	003300	003333	003399	000066	330066	650066
999000	993300	CC9900	006600	336666	0033FF	000099	660099	990066
CC0000	CC3300	FFCC00	009900	006666	0066FF	0000CC	663399	CC0099
FF0000	FF3300	FFFF00	00CC00	009999	0099FF	0000FF	9900CC	FF0099
CC3333	FF6600	FFFF33	00FF00	00CCCC	00CCFF	3366FF	9933FF	FF00FF
FF6666	FF6633	FFFF66	66FF66	66CCCC	00FFFF	3399FF	9966FF	FF66FF
FF9999	FF9966	FFFF99	99FF99	66FFCC	99FFFF	66CCFF	9999FF	FF99FF
FFCCCC	FFCC99	FFFFCC	CCFFCC	99FFCC	CCFFFF	99CCFF	CCCCFF	FFCCFF

Modificare una macro (9)

- Uso del debugger per vedere le operazioni passo-passo
- Ordinamento valori celle A2:A18

The screenshot shows the Microsoft Visual Basic Applications Edition interface. The main window displays the VBA code for a macro named 'SogliaSottolineaturaOrdinamento' in 'Modulo1'. The code is as follows:

```
' Scelta rapida da tastiera: CTRL+MAIUSC+C
Range("B1").Select
Application.CutCopyMode = False
ActiveCell.FormulaR1C1 = "Maggiore di 50"
Columns("B:B").Select
Selection.ColumnWidth = 17.27
Range("B1").Select
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .Color = 65535
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With
Range("A2:A18").Select
ActiveWorkbook.Worksheets("Foglio2").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("Foglio2").Sort.SortFields.Add Key:=Range("A2:A18") _
, SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
With ActiveWorkbook.Worksheets("Foglio2").Sort
    .SetRange Range("A1:A18")
    .Header = xlYes
    .MatchCase = False
    .Orientation = xlTopToBottom
    .SortMethod = xlPinYin
    .Apply
End With
Range("B2").Select
ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50, ""Maggiore"", ""Minore o uguale"")"
```

The left-hand pane shows the Excel spreadsheet with the following data in column A:

Valori	Maggiore di 50
1	
2	1
3	9
4	12
5	23
6	32
7	33
8	34
9	43
10	56
11	56
12	76
13	76
14	77
15	80
16	234
17	657
18	678
19	

Modificare una macro (10)

- Uso del debugger per vedere le operazioni passo-passo
- Fase finale: si scrive la formula nella B2 e si trascina B3:B18

Microsoft Visual Basic, Applications Edition - Esempio4.xlsx - [Modulo1 (codice)]

File Modifica Visualizza Inserisci Formato Debug Esegui Strumenti Aggiunte Finestra ?

Riga 49, Col 1

Progetto - VBAProject

VBAProject (Esempio4.xlsx)

- Microsoft Excel Oggetti
 - Foglio1 (Sheet1)
 - Foglio2 (Foglio1)
 - Foglio3 (Foglio2)
 - Foglio4 (Foglio3)
 - Questa_cartella_di_lavoro
- Moduli
 - Modulo1

Proprietà - Modulo1

Modulo1 Modulo

Alfabetico Per categoria

(Name) Modulo1

```
ActiveCell.FormulaR1C1 = "Maggiore di 50"
Columns("B:B").Select
Selection.ColumnWidth = 17.27
Range("B1").Select
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .Color = 65535
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With
Range("A2:A18").Select
ActiveWorkbook.Worksheets("Foglio2").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("Foglio2").Sort.SortFields.Add Key:=Range("A2:A18")
, SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
With ActiveWorkbook.Worksheets("Foglio2").Sort
    .SetRange Range("A1:A18")
    .Header = xlYes
    .MatchCase = False
    .Orientation = xlTopToBottom
    .SortMethod = xlPinYin
    .Apply
End With
Range("B2").Select
ActiveCell.FormulaR1C1 = "=IF(RC[-1]>50,""Maggiore"", ""Minore o uguale"")"
Range("B2").Select
Selection.AutoFill Destination:=Range("B2:B18"), Type:=xlFillDefault
Range("B2:B18").Select
End Sub
```

Scrivere una macro (1)

- Esempio Ciclo for

```
Sub CicloFor()
```

```
' Esempio di ciclo for
```

```
' Cells(x,y) --> Excel VBA considera la
```

```
' riga 1 e colonna 1
```

```
' esempio: Cells(1,3) è la cella C1
```

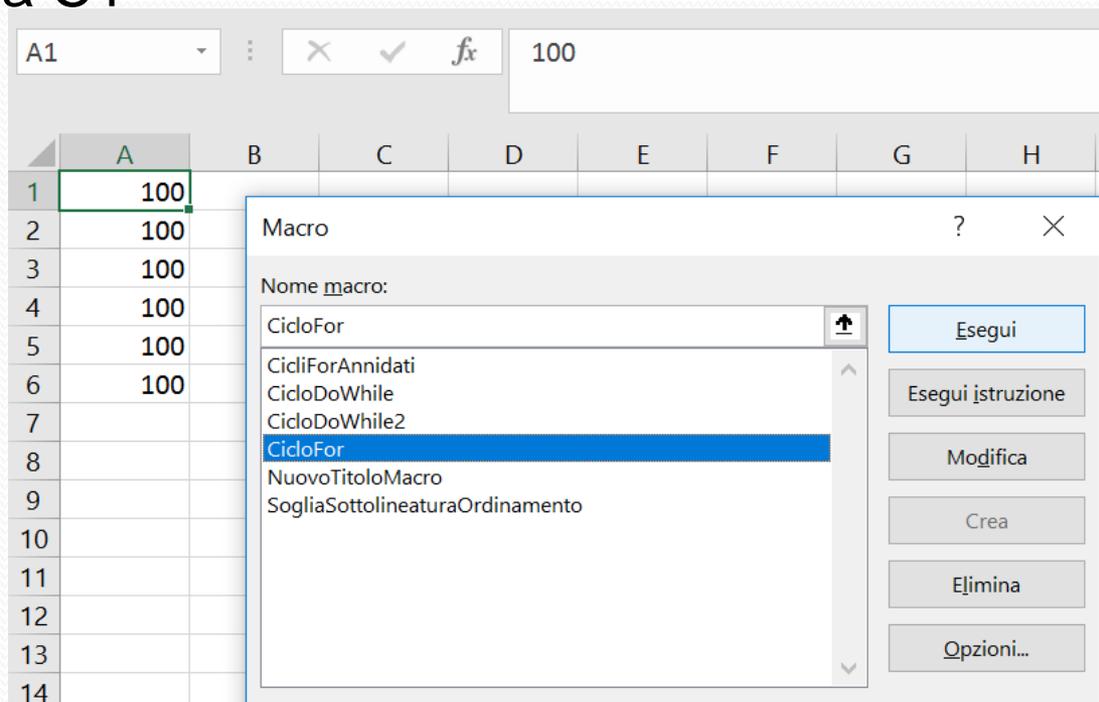
```
Dim i As Integer
```

```
For i = 1 To 6
```

```
    Cells(i, 1).Value = 100
```

```
Next i
```

```
End Sub
```



Scrivere una macro (2)

- Esempio Cicli for annidati:

```
Sub CicliForAnnidati()
```

```
' Esempio di cicli for annidati
```

```
' Cells(x,y) --> Excel VBA considera la
```

```
' riga 1 e colonna 1
```

```
' esempio: Cells(1,3) è la cella C1
```

```
Dim i As Integer, j As Integer
```

```
For i = 1 To 6
```

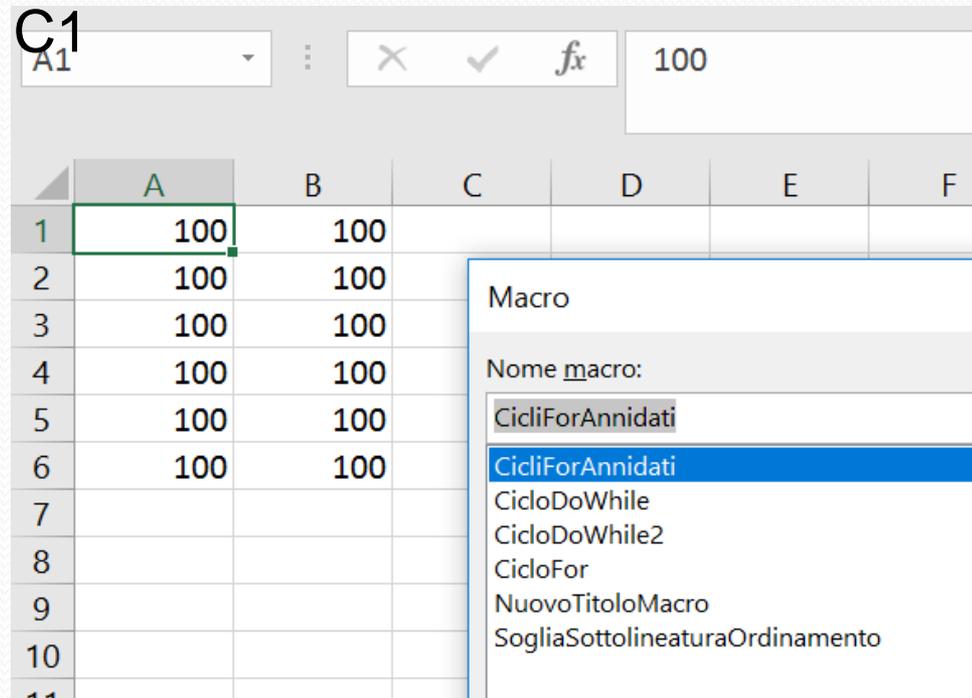
```
    For j = 1 To 2
```

```
        Cells(i, j).Value = 100
```

```
    Next j
```

```
Next i
```

```
End Sub
```



Scrivere una macro (3)

- Esempio Ciclo Do While:

```
Sub CicloDoWhile()
```

```
' Esempio di cicli for annidati
```

```
' Cells(x,y) --> Excel VBA considera la
```

```
' riga 1 e colonna 1
```

```
' esempio: Cells(1,3) è la cella C1
```

```
Dim i As Integer
```

```
i = 1
```

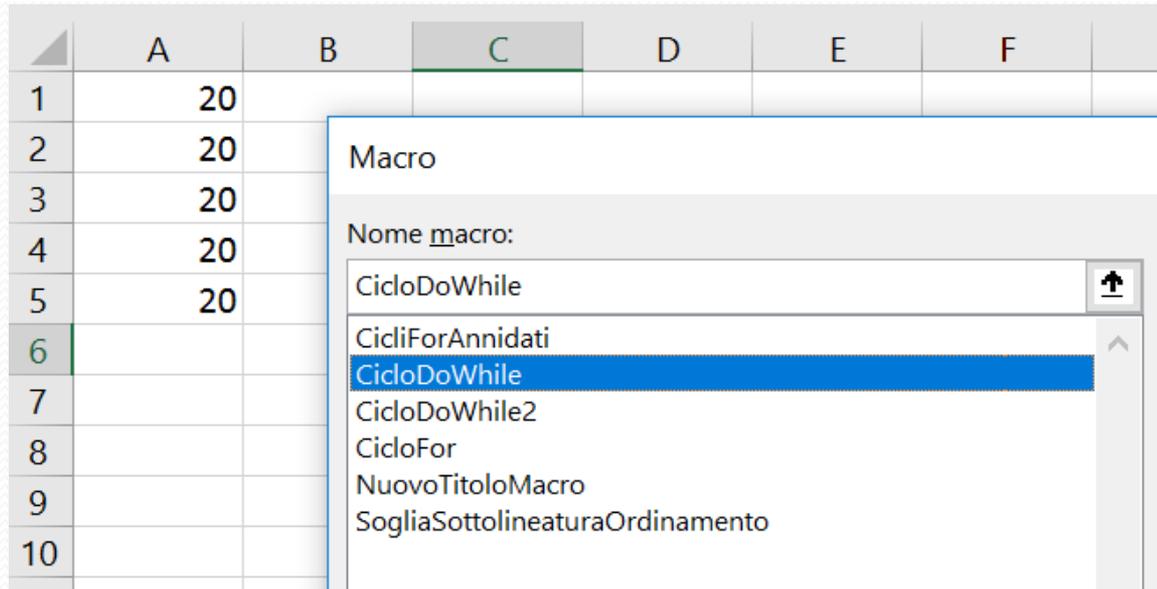
```
Do While i < 6
```

```
    Cells(i, 1).Value = 20
```

```
    i = i + 1
```

```
Loop
```

```
End Sub
```



Dal C a excel: scrivere un file da C e aprirlo da excel

- Si usano le funzioni viste in precedenza per la lettura e scrittura file da C
- Si scrivono i dati di output del programma in file csv
- I file csv contengono dati:
 - **In una riga separati da “;” o “,”**
 - **In una colonna separati dal carattere di “a capo”, ovvero in C nelle stringhe si inserisce il “\n”**

Esempio 1

```
#include <stdio.h>
main() {
    int i, num = 10;
    FILE *hand;
    float x, y;
    hand = fopen("dati.csv", "w+");//sovrascrivo il file se esiste
    if (!hand)
        printf("Impossibile aprire il file");
    else {
        fprintf(hand, "Scrivo: %d Colonne\n", num);
        for (i = 1; i <= num; i++) {
            fprintf(hand, "%d;", i); //1 riga con num (10) colonne
        }
        fclose(hand);
        printf("File aperto in scrittura! ... e chiuso subito dopo!\n");
    }
}
```

Esempio 1 – File csv

C:\WINDOWS\system32\cmd.exe

File aperto in scrittura! ... e chiuso subito dopo!
Premere un tasto per continuare . . .

Visual Studio 2015 > Projects > PrimoProgramma > PrimoProgramma

dati.csv - Excel

	A	B	C	D	E	F	G	H	I	J	K
1	Scrivo: 10 Colonne										
2	1	2	3	4	5	6	7	8	9	10	
3											

dati.csv - Blocco note

File Modifica Formato Visualizza ?

```
Scrivo: 10 Colonne  
1;2;3;4;5;6;7;8;9;10;
```

```
#include <stdio.h>
```

```
main() {
```

```
    int i, num = 10;
```

```
    FILE *hand;
```

```
    float x, y;
```

```
    hand = fopen("dati.csv", "w+"); //sovrascrivo il file se esiste
```

```
    if (!hand)
```

```
        printf("Impossibile aprire il file");
```

```
    else {
```

```
        fprintf(hand, "Colonna A\n di %d Righe\n", num); //header
```

```
        for (i = 1; i <= num; i++) {
```

```
            fprintf(hand, "%d\n", i); //1 Colonna e num (10) righe
```

```
        }
```

```
        fclose(hand);
```

```
        printf("File aperto in scrittura! ... e chiuso subito dopo!\n");
```

```
    }
```

```
}
```

Esempio 2

Esempio 2 – file csv

The screenshot illustrates a workflow for creating a CSV file. It shows a Windows command prompt window with the error message: "File aperto in scrittura! ... e chiuso subito dopo! Premere un tasto per continuare . . .". In the background, Visual Studio 2015 is open with a project named "PrimoProgramma". The file explorer shows a file named "dati.csv". An Excel spreadsheet titled "dati.csv - Excel" is open, displaying a table with 10 rows and 4 columns. The first row is labeled "Colonna A di 10 Righe". The second row contains the number 1, and the subsequent rows contain numbers 2 through 10. A Notepad window titled "dati.csv - Bl..." is also open, showing the same 10 rows of numbers, indicating that the data from the Excel spreadsheet has been copied into the text editor.

A	B	C	D
Colonna A di 10 Righe			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

dati.csv - Bl...
File Modifica Formato Visualizza ?
Colonna A di 10 Righe
1
2
3
4
5
6
7
8
9
10

Esempio 3

```
main() {  
int i, num = 10;  
FILE *hand;  
float x, y;  
hand = fopen("dati.csv", "w+");//sovrascrivo il file se esiste  
if (!hand)  
printf("Impossibile aprire il file");  
else {  
    fprintf(hand, "Scrivo: %d Righe\n", num);  
    fprintf(hand, "Colonna A;ColonnaB\n"); //header  
    int sum = 0;  
    for (i = 1; i <= num; i++) {  
        sum = sum + 10;  
        fprintf(hand, "%d;%d\n", i, sum);//2 colonne con num (10) righe  
    }  
    fclose(hand);  
    printf("File aperto in scrittura! ... e chiuso subito dopo!\n");  
}
```

Esempio 3 – file csv

The screenshot illustrates the process of creating and editing a CSV file. It shows a Windows command prompt window with the text "File aperto in scrittura! ... e chiuso subito dopo! Premere un tasto per continuare . . .". Below it, the Visual Studio 2015 interface is visible, showing the file explorer with the "dati.csv" file selected. The Excel application is open, displaying the "dati.csv" file with the following data:

	A	B	C	D
1	Scrive: 10 Righe			
2	Colonna A	ColonnaB		
3	1	10		
4	2	20		
5	3	30		
6	4	40		
7	5	50		
8	6	60		
9	7	70		
10	8	80		
11	9	90		
12	10	100		

The Notepad window, titled "dati.csv - Blocco note", shows the raw text content of the CSV file:

```
Scrive: 10 Righe  
Colonna A;ColonnaB  
1;10  
2;20  
3;30  
4;40  
5;50  
6;60  
7;70  
8;80  
9;90  
10;100
```

Link Utili

- Link Utili:
 - <http://www.disit.org>
 - <https://support.office.com>

Compiti



Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2016/17
Prof. Michela Paolucci -- 15/06/2017

1. **(10 punti)** Dato un array statico scrivere una funzione che:
 - a. Stampa a video gli elementi pari.
 - b. Conti e ritorni il numero degli elementi dispari.

2. **(20 punti)** Data una lista in forma collegata con puntatori scrivere una funzione che:
 - a. Ritorni il numero degli elementi della lista.
 - b. Il massimo degli elementi della lista.

3. **(5 punti)** Dato il numero decimale 935:
 - a. Convertire il numero in binario mostrando lo svolgimento delle divisioni successive.
 - b. Convertire il numero in esadecimale mostrando i passi della conversione.

Esercizio 1 (10 punti)

- Dato un array statico scrivere una funzione che:
 - i) stampi a video gli elementi pari
 - ii) conti e ritorni il numero degli elementi dispari

//Definizione funzione

```
int pariDispari(int *A, int n){
    int i, pari, count = 0;
    for (i = 0; i < n; i++) {
        pari = A[i] % 2; //Se il resto della divisione per 2 è 0, A[i] è pari
        if (pari==0)
            printf("Valore pari: A[%d]: %d\n", i, A[i]);
        else //se non è pari, allora è dispari e conto
            count++;
    }
    return count;
}
```

Esercizio 1: Esempio di uso della funzione

```
int pariDispari(int * A, int n);
```

```
main() {  
    int B[] = { 2, 0, 23, 53, 6, 3, 4, 44 };  
    int contaDispari, num;  
    num = sizeof(B) / sizeof(B[0]); //oppure con #define N 8  
    printf("Numero elementi: %d\n", num);  
    contaDispari = pariDispari(B, num);  
    printf("Numero di elementi dispari: %d\n", contaDispari);  
}
```

//Definizione funzione

```
int pariDispari(int *A, int n){...}
```

Esercizio 2 (20 punti) – Quesito 2a)

Università di Firenze -- Facoltà di Ingegneria

Corso di laurea in Ingegneria Meccanica e Gestionale (M-Z)

Prova Scritta di **Fondamenti di Informatica** A.A. 2016/17

Prof. Federico Pernici -- 15/06/2017

Prof. Michela Paolucci

- (10 punti)** Dato un array statico scrivere una funzione che:
 - Stampa a video gli elementi pari.
 - Conti e ritorni il numero degli elementi dispari.
- (20 punti)** Data una lista in forma collegata con puntatori scrivere una funzione che:
 - Ritorni il numero degli elementi della lista.
 - Il massimo degli elementi della lista.
- (5 punti)** Dato il numero decimale 935:
 - Convertire il numero in binario mostrando lo svolgimento delle divisioni successive.
 - Convertire il numero in esadecimale mostrando i passi della conversione.

Quesito 1a:

- *Si visita la lista*
- *Si definisce una variabile per il conteggio degli elementi.*
- *All'interno del ciclo di visita si incrementa la variabile di conteggio .*
- *Conclusa la visita si ritorna il valore del numero degli elementi.*
- *Nel main la funzione viene chiamata come:*

```
int nElem = countElemList(myList1);
```

```
int countElemList(struct intList * ptr){  
  
    int counter = 0;  
    while(ptr!=NULL){  
        counter++;  
        ptr=ptr->next_ptr;  
    }  
    return counter;  
}
```

Esercizio 2 (20 punti) – Quesito 2b)

Università di Firenze -- Facoltà di Ingegneria

Corso di laurea in Ingegneria Meccanica e Gestionale (M-Z)

Prova Scritta di **Fondamenti di Informatica** A.A. 2016/17

Prof. Federico Pernici -- 15/06/2017

Prof. Michela Paolucci

1. **(10 punti)** Dato un array statico scrivere una funzione che:
 - a. Stampa a video gli elementi pari.
 - b. Conti e ritorni il numero degli elementi dispari.
2. **(20 punti)** Data una lista in forma collegata con puntatori scrivere una funzione che:
 - a. Ritorni il numero degli elementi della lista.
 - b. Il massimo degli elementi della lista.
3. **(5 punti)** Dato il numero decimale 935:
 - a. Convertire il numero in binario mostrando lo svolgimento delle divisioni successive.
 - b. Convertire il numero in esadecimale mostrando i passi della conversione.

Quesito 2b:

- *Si visita la lista*
- *Si definisce un massimo corrente arbitrariamente piccolo.*
- *All'interno del ciclo di visita si controlla se l'elemento corrente della lista è maggiore del massimo corrente.*
- *Se la condizione è verificata si aggiorna il massimo corrente con il valore dell'elemento corrente.*
- *Conclusa la visita si ritorna il valore del massimo corrente.*
- *Nel main la funzione viene chiamata come:*

```
int maxVal = maxElemList(myList1);
```

```
int maxElemList(struct intList * ptr){  
    int currentMax = -999999999;  
    while(ptr!=NULL){  
        if (ptr->value > currentMax){  
            currentMax = ptr->value;  
        }  
        ptr=ptr->next_ptr;  
    }  
    return currentMax;  
}
```

Esercizio 3 – quesito a)

- Dato il numero decimale 935:
 - a) Convertire il numero in binario mostrando lo svolgimento delle divisioni successive;
 - b) Convertire il numero in esadecimale mostrando i passi della conversione

<i>Base 10</i>		<i>Resto della divisione per due</i>
935		1
467		1
233		1
116		0
58		0
29		1
14		0
7		1
3		1
1		1
0		0

$[935]_{10} = [1110100111]_2$

Esercizio 3 – quesito b)

- Dato il numero decimale 935:
 - a) Convertire il numero in binario mostrando lo svolgimento delle divisioni successive;
 - b) Convertire il numero in esadecimale mostrando i passi della conversione

Si parte dal risultato del quesito 3a):

$$[935]_{10} = [1110100111]_2$$

Si 'impaccano' i bit a partire da destra:

$$[0111]_2 = [0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0]_2 = [0+4+2+1]_2 = [7]_2 = [7]_{16}$$

$$[1010]_2 = [1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0]_2 = [8+0+2+0]_2 = [10]_2 = [A]_{16}$$

$$[0011]_2 = [0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0]_2 = [0+0+2+1]_2 = [3]_2 = [3]_{16}$$

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2016/17
Prof. Michela Paolucci -- 29/06/2017

1. **(10 punti)** Data una matrice scrivere una funzione che:
 - a. Calcoli e ritorni la somma degli elementi.
 - b. Calcoli e ritorni la media della i-esima riga.

2. **(20 punti)** Data una lista in forma collegata con puntatori scrivere:
 - a. Una funzione che prenda in ingresso un valore intero e calcoli il numero di occorrenze di tale valore nella lista
 - b. Una funzione che prenda in ingresso due liste, ne effettui il confronto e determini se tali liste sono uguali o meno.

3. **(5 punti)** Dato il numero $A=564$:
 - a. Convertire il numero in binario mostrando lo svolgimento delle divisioni successive
 - b. Calcolare il complemento a 2 del risultato del punto precedente

Esercizio 1 (10 punti)

- Data una matrice scrivere una funzione che:
 - i) calcoli e ritorni la somma degli elementi
 - ii) calcoli e ritorni la media della i-esima riga

//Definizione funzione punto i)

```
int somma(int *B, int r, int c) {  
    int sum = 0, i, j;  
    for (i = 0; i < r; i++) { //ciclo sulle righe  
        for (j = 0; j < c; j++) { //ciclo sulle colonne  
            sum = sum + B[c*i + j];  
        }  
    }  
    return sum;  
}
```

Esercizio 1 (10 punti)

- Data una matrice scrivere una funzione che:
 - i) calcoli e ritorni la somma degli elementi
 - ii) calcoli e ritorni la media della i-esima riga

//Definizione funzione punto ii)

```
float media(int *B, int r, int c, int R) {
    int sum = 0, i, j;
    float media = 0;
    for (i = 0; i < r; i++) { //ciclo sulle righe
        if(i==R){
            for (j = 0; j < c; j++) { //ciclo sulle colonne
                sum = sum + B[c*i + j];
            }
            media = (float)sum / c;
        }
    }
    return media;
}
```

Esercizio 1 (10 punti)

- Data una matrice scrivere una funzione che:
 - i) calcoli e ritorni la somma degli elementi
 - ii) calcoli e ritorni la media della i-esima riga

//Definizione funzione punto ii) [alternativa punto ii)]

```
float media(int *B, int r, int c, int R) {  
    int sum = 0, j;  
    for (j = 0; j < c; j++) {  
        sum = sum + B[c*R + j];  
    }  
    return (float)sum / c;  
}
```

Esercizio 1: Esempio di uso delle funzioni

```
#define NR 3
#define NC 4
int somma(int *B, int r, int c);
float media(int *B, int r, int c, int R);

main() {
    int totale;
    int B[NR][NC] = { { 0,10,0,0 }, { 1,1,1,1 }, { 1,1,1,1 } };
    totale = somma(B, NR, NC);
    printf("Totale %d", totale);
    float med;
    med = media(B, NR, NC, 0);
    printf("media della riga %d: %f", 0, med);
}
```

Esercizio 1 (10 punti) – [alternativa i e ii]

```
//funzione punto i)
#define NR 3
#define NC 4
//dichiarazione
int somma2(int A[NR][NC]);
//definizione
int somma2(int A[NR][NC]){
    int sum = 0, i, j;
    for (i = 0; i < NR; i++) { //ciclo sulle righe
        for (j = 0; j < NC; j++) { //ciclo sulle colonne
            sum = sum + A[i][j];
        }
    }
    return sum;
}
```

- Data una matrice scrivere una funzione che:
 - i) calcoli e ritorni la somma degli elementi
 - ii) calcoli e ritorni la media della i-esima riga

Esercizio 1 (10 punti) - alternativa

//funzione punto ii)

```
#define NR 3
```

```
#define NC 4
```

```
int media2(int A[NR][NC]); //dichiarazione
```

```
float media2(int A[NR][NC], int R) { //definizione
```

```
    int sum = 0, i, j;
```

```
    float media;
```

```
    for (i = 0; i < NR; i++) { //ciclo sulle righe
```

```
        if(i==R){
```

```
            for (j = 0; j < NC; j++) { //ciclo sulle colonne
```

```
                sum = sum + A[i][j];
```

```
            }
```

```
            media = (float)sum / NC;
```

```
        }
```

```
    }
```

```
    return media;
```

- Data una matrice scrivere una funzione che:
 - i) calcoli e ritorni la somma degli elementi
 - ii) calcoli e ritorni la media della i-esima riga

Esercizio 2 (20 punti) – Quesito 2a)

2. (20 punti) Data una lista in forma collegata con puntatori, scrivere:
- Una funzione che prenda in ingresso un valore intero e calcoli il numero di occorrenze di tale valore nella lista.
 - Una funzione che prenda in ingresso due liste, ne effettui il confronto e determini se tali liste sono uguali o meno.

Quesito 2a:

- *Si visita la lista*
- *Si controlla se il valore passato è uguale all'elemento corrente della lista.*
- *In caso affermativo si incrementa una variabile di conteggio.*
- *Conclusa la visita si ritorna il valore del numero degli elementi.*
- *Nel main la funzione viene chiamata come:*

```
int nOcc = numOccurringElem(myList1, val);
```

```
int numOccurringElem(struct intList * ptr, int val){
int count=0;
while(ptr!=NULL){
    if (ptr->value == val){
        count++;
    }
    ptr=ptr->next_ptr;
}
return count;
}
```

Esercizio 2 (20 punti) – Quesito 2b)

2. (20 punti) Data una lista in forma collegata con puntatori, scrivere:
- Una funzione che prenda in ingresso un valore intero e calcoli il numero di occorrenze di tale valore nella lista.
 - Una funzione che prenda in ingresso due liste, ne effettui il confronto e determini se tali liste sono uguali o meno.

Quesito 2b:

- Si visitano simultaneamente le due liste.
- Si controlla se i rispettivi elementi correnti sono diversi. In caso affermativo si esce dalla funzione e si ritorna esito negativo (i.e. le liste sono diverse).
- Se la visita simultanea si conclude si controlla se entrambe le liste terminano con un NULL. In caso affermativo si ritorna esito positivo (i.e. sono uguali).
- Nel main la funzione viene chiamata come:

```
Boolean testEqual = compareList(myList1, myList2);
```

```
Boolean compareList(struct intList * ptr1, struct intList * ptr2){
```

```
while(ptr1!=NULL && ptr2!=NULL){
    if (ptr1->value != ptr2->value){
        return FALSE;
    }
    ptr1=ptr1->next_ptr;
    ptr2=ptr2->next_ptr;
}
if (ptr1 == NULL && ptr2 == NULL){
    return TRUE;
} else {
    return FALSE;
}
}
```

Esercizio 3 – quesito a)

- Dato il numero decimale $A=564$:
 - a) Convertire il numero in binario mostrando lo svolgimento delle divisioni successive;
 - b) Calcolare il complemento a 2 del risultato del punto a)

<i>Base 10</i>		<i>Resto della divisione per due</i>
564		0
282		0
141		1
70		0
35		1
17		1
8		0
4		0
2		0
1		1
0		0

$[564]_{10} = [1000110100]_2$

Esercizio 3 – quesito b)

- Dato il numero decimale $A=564$:
 - a) Convertire il numero in binario mostrando lo svolgimento delle divisioni successive;
 - b) Calcolare il complemento a 2 del risultato del punto a)

Si parte dal risultato del quesito 3a):

$$[564]_{10} = [1000110100]_2$$

Si invertono i bit a partire dal primo 1 meno significativo NON compreso:

$$A = 1000110100$$

$$C_2(A) = 0111001100$$

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2016/17
Prof. Michela Paolucci - 13/07/2017

1. **(10 punti)** Data una matrice statica scrivere **una sola** funzione che:
 - a. Calcoli e ritorni il valor medio degli elementi.
 - b. Stampi a video il valore degli elementi che hanno indice di riga uguale all'indice della colonna.

2. **(20 punti)** Data una lista di float, in forma collegata con puntatori:
 - a. Scrivere una funzione che ritorni la lunghezza di tale lista.
 - b. Scrivere una funzione che, data una soglia in input, calcoli la somma dei valori maggiori e uguali rispetto a tale soglia

3. **(5 punti)** Dati i numeri decimali $A=823$, $B=176$:
 - a. Convertire i numeri in binario mostrando lo svolgimento
 - b. Calcolare la somma in base 2 tra i due numeri ($[A]_2 + [B]_2$)

Esercizio 1 (10 punti) - testo

- Data una matrice statica scrivere **una sola funzione** che:
 - Calcoli e ritorni il valor medio degli elementi
 - Stampi a video il valore degli elementi che hanno indice di riga uguale all'indice della colonna.

Esercizio 1 (10 punti)

- **Quesito 1a:**
- Si visita la matrice mediante due cicli nidificati e si sommano gli elementi.
- Si ritorna la media e si stampano gli elementi quando $i==j$

```
float meanMatrix(int *B, int r, int c) {  
    int sum = 0, j, i;  
    for (i = 0; i < r; i++) {  
        for (j = 0; j < c; j++) {  
            sum = sum + B[i*r + j];  
            if (i==j) {  
                printf("%i", B[i*r + j]);  
            }  
        }  
    }  
    return (float)sum / (r*c);  
}
```

Esercizio 2 (20 punti) - testo

- **(20 punti)** Data una lista di float, in forma collegata con puntatori:
 - a) Scrivere una funzione che ritorni la lunghezza di tale lista
 - b) Scrivere una funzione che, data una soglia in input, calcoli la somma dei valori maggiori e uguali rispetto a tale soglia

Esercizio 2 (20 punti) – Quesito a)

- **Quesito 2a:**
 - *Si visita la lista*
 - *Si definisce una variabile per il conteggio degli elementi*
 - *All'interno del ciclo di visita si incrementa la variabile di conteggio*
 - *Conclusa la visita si ritorna il valore del numero degli elementi*
 - *Nel main la funzione viene chiamata come:*
 - `int len = lengthList(myList1);`

```
int lengthList(struct intList * ptr){  
    int counter = 0;  
    while(ptr!=NULL){  
        counter++;  
        ptr=ptr->next_ptr;  
    }  
    return counter;  
}
```

Esercizio 2 (20 punti) – Quesito b)

Quesito 2b:

- Si visita la lista
- Si controlla se il valore passato è maggiore uguale all'elemento corrente della lista
- In caso affermativo si incrementa una variabile di conteggio
- Conclusa la visita si ritorna il valore del numero degli elementi
- Nel main la funzione viene chiamata come:

```
int sum = sumLargerThanThresh(myList1, val);
```

```
int sumLargerThanThresh(struct intList * ptr, int
val){
    int sum=0;
    while(ptr!=NULL){
        if (ptr->value >= val){
            sum = sum + ptr->value;
        }
        ptr=ptr->next_ptr;
    }
    return sum;
}
```

Esercizio 3 (5 punti) – testo

- Dati i numeri decimali $A=823$, $B=176$:
 - Convertire i numeri in binario mostrando lo svolgimento
 - Calcolare la differenza in base 2 tra i due numeri ($[A]_2 + [B]_2$)

Esercizio 3 – parte i)

Base 10	Resto della divisione per due		Base 10	Resto divisione per 2
823	1	$[823]_{10} = [1100110111]_2$ $[176]_{10} = [10110000]_2$	176	0
411	1		88	0
205	1		44	0
102	0		22	0
51	1		11	1
25	1		5	1
12	0		2	0
6	0		1	1
3	1		0	0
1	1			
0	0			

1 1

1 1 0 0 1 1 0 1 1 1

0 0 1 0 1 1 0 0 0 0

1 1 1 1 1 0 0 1 1 1

$[176+823]_{10} = [999]_{10} =$
 $[1111100111]_2$

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2016/17
Prof. Michela Paolucci - 14/09/2017

1. **(6 punti)** Dato un albero binario di ricerca a valori interi scrivere:
 - a. Una funzione che ritorni il numero totale dei nodi con valore pari

2. **(18 punti)** Data una lista di interi, in forma collegata con puntatori, scrivere:
 - a. Una funzione che calcoli le occorrenze di ciascun elemento e le stampi a video
 - b. Una funzione che restituisca la somma dei valori divisibili per 3 presenti nella lista

3. **(4 punti)** Data una matrice statica scrivere **una** funzione che:
 - a. Restituisca la somma dei valori degli elementi che hanno indice della riga maggiore dell'indice della colonna

4. **(2 punti)** Dati i numeri decimali $A=54$, $B=123$:
 - a. Convertire i numeri in binario mostrando lo svolgimento

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2016/17
Prof. Michela Paolucci - 14/09/2017

1. **(6 punti)** Dato un albero binario di ricerca a valori interi scrivere:
 - a. Una funzione che ritorni il numero totale dei nodi con valore pari

2. **(18 punti)** Data una lista di interi, in forma collegata con puntatori, scrivere:
 - a. Una funzione che calcoli le occorrenze di ciascun elemento e le stampi a video
 - b. Una funzione che restituisca la somma dei valori divisibili per 3 presenti nella lista

3. **(4 punti)** Data una matrice statica scrivere **una** funzione che:
 - a. Restituisca la somma dei valori degli elementi che hanno indice della riga maggiore dell'indice della colonna

4. **(2 punti)** Dati i numeri decimali $A=54$, $B=123$:
 - a. Convertire i numeri in binario mostrando lo svolgimento

Esercizio 1 (6 punti)

Quesito 1:

- Si parte dall'attraversamento ricorsivo di un'albero binario di ricerca.
- Si sostituisce alla stampa del nodo il codice di conteggio nel caso in cui l'elemento è di valore pari.
- Il puntatore al contatore viene passato ad ogni chiamata ricorsiva.
- Nel main la funzione viene chiamata come:

```
int numElem = 0;  
countTreeEven(ptrTree, &numElem);
```

```
void countTreeEven(struct btree * ptrTree, int * nTreeElem) {  
    if(ptrTree!=NULL) {  
        countTreeLeaf(ptrTree->leftptr, nTreeElem );  
        if (ptrTree->value % 2 == 0){  
            *nTreeElem = *nTreeElem + 1;  
        }  
        countTreeLeaf(ptrTree->rightptr, nTreeElem );  
    }  
}
```

Esercizio 2 – Quesito 2a (9 punti)

- Si visita la lista e si determinano le occorrenze per ogni elemento visitato.
- Si definisce una funzione che calcola il numero delle occorrenze per un elemento dato (*numOccurringElem*).
- La funzione di stampa chiama *numOccurringElem* per ciascun elemento visitato.
- Per la visita si utilizza un puntatore di appoggio *ptr1* per non perdere il puntatore di inizio lista.

```
void printListOccur(struct intList * myList1){
    int occur;
    struct intList* ptr1 = myList1;
    while (ptr1 != NULL){
        occur = numOccurringElem(myList1,ptr1->value);
        printf("%i ha %i occorrenze\n",ptr1->value,occur);
        ptr1 = ptr1->next_ptr;
    }
}
```

```
int numOccurringElem(struct intList * ptr, int val){
    int count=0;
    while(ptr!=NULL){
        if (ptr->value == val){
            count++;
        }
        ptr=ptr->next_ptr;
    }
    return count;
}
```

Esercizio 2 – Quesito 2b (9 punti)

- *Si visita la lista*
- *Si controlla se il valore passato è divisibile per 3.*
- *In caso affermativo somma l'elemento.*
- *Conclusa la visita si ritorna il valore della somma degli elementi trovati.*

```
int sumDivisibleBy3(struct intList * ptr, int val){
    int sum=0;
    while(ptr!=NULL){
        if (ptr->value % 3 == 0){
            sum = sum + ptr->value;
        }
        ptr=ptr->next_ptr;
    }
    return sum;
}
```

Esercizio 3 (4 punti)

- *Si effettuano cicli sulla matrice*
- *Si controlla se l'indice della riga è maggiore dell'indice della colonna*
- *In caso affermativo si somma l'elemento.*
- *Conclusa la vista si ritorna il valore della somma*

```
int sum_under_diag(int *B, int r, int c) {
    int sum = 0, i, j;
    for (i = 0; i < r; i++) { //ciclo sulle righe
        for (j = 0; j < c; j++) { //ciclo sulle colonne
            if(i > j)
                sum = sum + B[c* i + j ];
        }
    }
    return sum;
}
```

Esercizio 3 (2 punti)

Base 10	Resto della divisione per due
54	0
27	1
13	1
6	0
3	1
1	1
0	0

$[54]_{10} = [110110]_2$

Base 10	Resto della divisione per due
123	1
61	1
30	0
15	1
7	1
3	1
1	1
0	0

$[123]_{10} = [1111011]_2$

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2016/17
Prof. Michela Paolucci - 16/11/2017

Tempo: 2 ore

1. **(12 punti)** Data una matrice scrivere:
 - a. Una funzione che calcoli e ritorni la somma degli elementi della j -esima colonna.
 - b. Una funzione che stampi gli indici di riga e colonna dell'elemento con valore minimo.

1. **(18 punti)** Date due liste di valori a interi A e B della stessa lunghezza, scrivere:
 - a. Una funzione che ritorna la somma $A+B$ in una terza lista C.
 - b. Una funzione che ritorna il prodotto $A*B$ nella lista B.

Quesito 1 (12 punti)

Quesito 1b:

- Si visita la matrice mediante due cicli nidificati e si sommano gli elementi.
- Si aggiorna il minimo solo se il valore attuale è minore di esso

```
void int_indici_min(int *A, int r, int c) {
    int min = 999999999;
    int i, j;
    int J=0, I=0;
    for (i = 0; i < r; i++) {//ciclo sulle righe
        for (j = 0; j < c; j++) {//ciclo sulle colonne
            if (A[c*i + j] < min){
                min = A[c*i + j];
                I = i;
                J = j;
            }
        }
    }
    printf("Elemento minimo: %d, con indice di riga: %d e
        indice di colonna:%d", min, I, J);
}
```

Quesito 1 (12 punti)

Quesito 1a:

- Si visita la matrice mediante due cicli nidificati e si sommano gli elementi.
- Si ritorna la somma e si stampano gli elementi quando $j==J$

```
int sum_diag(int *B, int r, int c, int J) {
    int sum = 0, i, j;
    for (i = 0; i < r; i++) { //ciclo sulle righe
        for (j = 0; j < c; j++) { //ciclo sulle colonne
            if (i == J)
                sum = sum + B[c*i + j];
        }
    }
    return sum;
}
```

Quesito 2 (18 punti)

Quesito 2a:

- *Si visitano le due liste contemporaneamente*
- *Si inserisce la somma degli elementi visitati in una terza lista*
- *Si ritorna il puntatore della terza lista.*

```
struct list * SumTwoList(struct list * ptr1, struct list * ptr2){
struct list * ptr3;
init(&ptr3);
while(ptr1!=NULL){
    suf_insert(&ptr3 , ptr1->value + ptr2->value);
    ptr1=ptr1->next_ptr;
    ptr2=ptr2->next_ptr;
}
return ptr3;
}
```

Quesito 2 (18 punti)

Quesito 2b:

- *Si visitano le due liste contemporaneamente*
- *Si sovrascrive la somma delle due liste sulla prima.*

```
void SumToList(struct list * ptr1, struct list * ptr2){  
    while(ptr1!=NULL){  
        ptr1->value= ptr1->value + ptr2->value);  
        ptr1=ptr1->next_ptr;  
        ptr2=ptr2->next_ptr;  
    }  
}
```

Università di Firenze - Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)

Prova Scritta di **Fondamenti di Informatica**

Prof. Michela Paolucci - 18/01/2018

- (10 punti)** Dato un array statico A e uno dinamico B delle stesse dimensioni, scrivere una funzione che:
 - Ritorna un terzo array C contenente il prodotto per componenti $C=A*B$.
 - Ritorna il rapporto per componenti dei due array nell'array B.
- (15 punti)** Data una lista di valori interi scrivere una funzione che:
 - Inserisce il valore 0 in posizione antecedente agli elementi contenenti valori pari.
 - Inserisce il valore 0 in posizione successiva agli elementi contenenti valori dispari.
- (5 punti)** Dato un albero binario di ricerca:
 - Scrivere una funzione che stampi i soli nodi che hanno solo il figlio destro.

Quesito 1 (10 punti)

Quesito 1a e 1b:

- Si passano entrambi gli array e la loro lunghezza.
- Per il punto a. si alloca un nuovo array dinamicamente (D) e si inseriscono i valori del prodotto $A*B$. La memoria allocata dinamicamente rimane accessibile fuori dalla funzione mediante il puntatore ritornato.
- Per il punto b. si sovrascrivono in B i valori del rapporto B/A . Il vettore è già allocato fuori dalla funzione.

```
int * arrayFunction(int* A, int* B, int N){
    int *D = (int *)calloc(N, sizeof(int));
    for ( int i = 0; i < N; i++ ){// a.
        D[i]=A[i]*B[i];
    }
    for ( int i = 0; i < N; i++ ){// b.
        B[i]=B[i]/A[i];
    }
    return d;
}
```

Quesito 2 (15 punti)

```
void pari_dispari_insert(struct List **ptr) {
    struct List *tmp_ptr; // lista di appoggio
    init(&tmp_ptr); // ricordo: void init(struct List ** ptrptr)
    tmp_ptr = *ptr;
    *ptr = NULL;
    while (tmp_ptr != NULL) {
        if (tmp_ptr->value == 0)
            suf_insert(ptr, 0); // lo inserisco
        else {
            if (tmp_ptr->value % 2 == 0) { // è pari
                suf_insert(ptr, 0);
                suf_insert(ptr, tmp_ptr->value);
            } // void suf_insert(struct List ** ptrptr, int value);
            else {
                suf_insert(ptr, tmp_ptr->value);
                suf_insert(ptr, 0);
            } // void pre_insert(struct List ** ptrptr, int value);
        }
        tmp_ptr = tmp_ptr->next_ptr;
    }
}
```

Quesito 2 - dettagli

//struttura della lista in forma collegata con puntatori

```
struct list {  
int value;  
struct list * next_ptr;  
};
```

//dichiarazione

```
void pari_dispari_insert(struct list ** ptr);
```

```
main(){
```

//inizializzo la lista

```
struct list *lista;
```

```
init(&lista);
```

```
struct list **ptrlistaC, **ptrlista;
```

```
ptrlista = &lista;
```

[...]

```
pari_dispari_insert(ptrlista);
```

```
printf("Ho inserito gli zeri,
```

```
ora stampo la lista \n");
```

```
visit(lista); }
```

Quesito 3 (5 punti)

Quesito 3:

- Si parte dall'attraversamento ricorsivo di un'albero binario di ricerca.
- Si aggiunge alla stampa del nodo il codice per verificare i nodi con figlio dx e senza figlio sx.
- Nel main la funzione viene chiamata come:
`printTreeElemRight(ptrTree);`

```
void printTreeElemRight(struct btree * ptrTree) {  
  
    if(ptrTree!=NULL) {  
        printTreeElemRight(ptrTree->leftptr);  
        if ( ptrTree->leftptr == NULL && ptrTree->rightptr != NULL ){  
            printf("%f \n",ptrTree->value);  
        }  
        printTreeElemRight(ptrTree->rightptr);  
    }  
  
}
```

Università di Firenze - Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2016/17
Prof. Michela Paolucci - 22/02/2018

1. **(10 punti)** Date due matrici quadrate delle stesse dimensioni A e B, scrivere una funzione che:
 - a. Ritorna una terza matrice C contenente la somma per componenti $C=A+B$.
 - b. Ritorna un array contenente gli elementi della diagonale.

2. **(10 punti)** Date due liste A e B di lunghezza diversa scrivere una funzione che:
 - a. Concatena, una dopo l'altra, le due liste in una terza lista C.
 - b. Concatena alla lista A, la lista B.

3. **(10 punti)** Dato un albero binario di ricerca:
 - a. Scrivere una funzione che stampi i soli nodi che hanno il figlio sinistro e il cui valore è maggiore del valore della radice.

Quesito 1a:

- Si visitano simultaneamente le matrici A e B mediante due cicli nidificati e si sommano gli elementi.
- Si ritorna la matrice C mediante il suo puntatore.

```
float * sumTwoMatrices(float *A, float *B, int n) {  
  
float *C = (float *)calloc(n*n , sizeof(float));  
int j, i;  
for (i = 0; i < n; i++) {  
    for (j = 0; j < n; j++) {  
        C[i*n + j] = A[i*n + j] + B[i*n + j];  
    }  
}  
return C;  
}
```

Quesito 1b:

- Si visitano gli elementi della diagonale e si scrivono gli elementi in un array D.
- Si ritorna l'array D mediante il suo puntatore.

```
float * extractDiag(float *A, int n) {  
    float *D = (float *)calloc(n ,  
        sizeof(float));  
    int i;  
    for (i = 0; i < n; i++) {  
        D[i] = A[i*n + i] + B[i*n + i];  
    }  
    return D;  
}
```

Quesito 2a:

- Si attraversano una dopo l'altra le due liste.
- Si aggiungono gli elementi attraversati alla lista C.
- Si ritorna la lista C.

```
struct intList * ConcatABtoC (struct intList * A, struct
intList * B){
    struct intList * C;
    initIntList(&C);
    while(A!=NULL){
        suf_insertInt(&C , A->value);
        A=A->next_ptr;
    }
    while(B!=NULL){
        suf_insertInt(&C , B->value);
        B=B->next_ptr;
    }
    return C;
}
```

Quesito 2b:

- *Si attraversa la lista A e ci si «salva» l'indirizzo dell'elemento finale in una variabile d'appoggio.*
- *Il prossimo dell'elemento finale viene sostituito con il primo della lista B*

```
void ConcatABtoA(struct intList * A, struct intList * B){  
  
    struct intList * tmpPtr;  
  
    while(A!=NULL){  
        if (A->next_ptr==NULL) {  
            tmpPtr = A;  
        }  
        A=A->next_ptr;  
    }  
    tmpPtr->next_ptr = B;  
}
```

Quesito 3:

- Si parte dall'attraversamento ricorsivo di un'albero binario di ricerca.
- Si aggiunge alla stampa del nodo il codice per verificare i nodi con figlio sx.
- I nodi con valore maggiore della radice sono quelli del sottoalbero di destra della radice quindi la funzione viene chiamata come:

```
printTreeElemLeft(ptrTree->rightptr);
```

```
void printTreeElemLeft(struct btree * ptrTree) {  
  
    if(ptrTree!=NULL) {  
        printTreeElemLeft(ptrTree->leftptr);  
        if ( ptrTree->leftptr !=NULL){  
            printf("%f \n",ptrTree->value);  
        }  
        printTreeElemLeft(ptrTree->rightptr);  
    }  
}
```

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2017/18
Prof. Michela Paolucci - 28/06/2018

1. **(12 punti)** Date due matrici quadrate delle stesse dimensioni A e B, scrivere le seguenti funzioni:
 - a. Funzione che scrive zero sulle colonne dispari di A e di B.
 - b. Funzione che scambia le colonne pari di A e B fra di loro.
2. **(10 punti)** Data una lista in forma collegata con puntatori, scrivere le seguenti funzioni:
 - a. Funzione che ritorna una nuova lista con l'ordine degli elementi invertito.
 - b. Funzione che ritorna il massimo ed il minimo della lista.
3. **(8 punti)** Dato un albero binario di ricerca, scrivere una funzione che:
 - a. Stampa il valore contenuto nei nodi che hanno sia figlio destro che il figlio sinistro.

Quesito 1a:

- Si attraversano le matrici, si controlla che la colonna corrente sia dispari e si scrive zero.

```
void evenColSetZero(int * A, int * B, int n){ int i,j;
    for (i=0;i<n;i++){
        for (j=0;j<n;j++){
            if ((j)%2!=0){
                A[i*n+j] = 0;
                B[i*n+j] = 0;
            }
        }
    }
}
```

Quesito 1b:

- Si attraversano le matrici, si controlla che la colonna corrente sia pari e si scambiano gli elementi correnti.

```
void swapColMatrices(int * A, int * B, int n){
    int i,j,tmp;
    for (i=0;i<n;i++){
        for (j=0;j<n;j++){
            if ((j)%2==0){
                tmp = A[i*n+j];
                A[i*n+j] = B[i*n+j];
                B[i*n+j] = tmp;
            }
        }
    }
}
```

Quesito 2a:

- *Si visita la lista, inserendo gli elementi della vecchia in testa alla nuova.*

```
void invert(struct list * ptr, struct list ** ptrptr){
    init(ptrptr);
    while (ptr!=NULL){
        pre_insert(ptrptr,ptr->value);
        ptr=ptr->next_ptr;
    }
}
```

Quesito 2a:

- *Si visita la lista, inserendo gli elementi della vecchia in testa alla nuova.*

```
void invert(struct list * ptr, struct list ** ptrptr){
    init(ptrptr);
    while (ptr!=NULL){
        pre_insert(ptrptr,ptr->value);
        ptr=ptr->next_ptr;
    }
}
```

Quesito 2b:

- Si visita la lista, e si mantengono i valori correnti del minimo e del massimo.

```
void maxMinList(struct list * ptr, float * max, float * m
in){
float currentMax = -999999; float currentMin = +999999;
while(ptr!=NULL){
    if (ptr->value > currentMax){
        currentMax = ptr->value;
    }
    if (ptr->value < currentMin){
        currentMin = ptr->value;
    }
    ptr=ptr->next_ptr;
}
*max = currentMax;
*min = currentMin;
}
```

Quesito 3:

- Si visita l'albero, si verificano che entrambi figli siano nulli e nel caso si stampa l'elemento corrente.

```
void printTreeElemLeft(struct btree * ptrTree) {  
  
    if(ptrTree!=NULL) {  
  
        printTreeElemLeft(ptrTree->leftptr);  
  
        if ( ptrTree->leftptr != NULL && ptrTree->rightptr !=  
NULL){  
            printf("%f \n", ptrTree->value);  
        }  
  
        printTreeElemLeft(ptrTree->rightptr);  
    }  
  
}
```

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2017/18
Prof. Michela Paolucci - 12/07/2018

1. **(14 punti)** Data una matrice A scrivere una funzione che:
 - a. Ritorni una nuova matrice contenente la trasposta di A .
 - b. Funzione che ritorna i valori massimi di ogni riga oppure colonna.

2. **(10 punti)** Data una lista in forma collegata con puntatori, scrivere le seguenti funzioni:
 - a. Funzione che ritorna una nuova lista con il valore degli elementi raddoppiati.
 - b. Funzione che ritorna la somma e stampa la media.

3. **(6 punti)** Dato un albero binario di ricerca, scrivere una funzione che:
 - a. Stampare il valore contenuto nei nodi il cui valore è compreso nell'intervallo definito tra due valori e con solo figlio destro.

Quesito 1a:

- *Si attraversano le matrici, si invertono righe di A con colonne di B e viceversa.*

```
int * transpose(int *A, int r, int c) {
    int *B = (int *)malloc((r*c) *
sizeof(int));
    for (int i = 0; i < r; i++) {//ciclo
righe
        for (int j = 0; j < c; j++) {//ciclo
col
            B[r*j+ i] = A[c*i + j];
        }
    }
    return B;
}
```

Quesito 1b:

- *Si attraversa la matrice*
- *Per ogni riga aggiorno un massimo corrente.*
- *Copio il valore nell'array di uscita.*
- *Quando tutti sono stati copiati ritorno l'array.*

```
int * maxRowMatrix(int * A, int r, int c){
    int i,j,currentRowMax;
    // alloco vettore dei massimi
    int *maxAr = (int *)malloc( c * sizeof(int));
    // visito la matrice
    for (i=0;i<r;i++){
        currentRowMax = -999999;
        // se elemento riga maggiore max corrente...
        for (j=0;j<c;j++){
            if (A[i*r+j] > currentRowMax){
                currentRowMax = A[i*r+j];
            }
        }
        maxAr[i] = currentRowMax;
    }
    return maxAr; }
```

Quesito 2a:

- *Si visita la lista, inserendo gli elementi della vecchia moltiplicati per due, in coda alla nuova.*

```
void doubleValues(struct list *ptr, struct list **ptrptr) {  
    init(ptrptr);  
    while (ptr != NULL) {  
        suf_insert(ptrptr, 2*ptr->value);  
        ptr = ptr->next_ptr;  
    }  
}
```

Quesito 2b:

- *Si visita la lista*
- *Per ogni nodo visitato: i) si incrementa un contatore; ii) si aggiorna la somma dei volari*
- *Finita la visita dei nodi si stampa la media e si rende la somma*

```
int sumMean(struct List *ptr) {  
    int count = 0, sum=0;  
    float mean;  
    while (ptr != NULL) {  
        count++;  
        sum = sum + ptr->value;  
        ptr = ptr->next_ptr;  
    }  
    mean = (float)sum / count;  
    printf("\nMedia: %f", mean);  
    return sum;  
}
```

Quesito 3:

- *Si visita l'albero, si verificano che entrambi figli siano nulli e nel caso si stampa l'elemento corrente.*

```
void printTreeElemSoglie(struct btree * ptrTree,
                        float soglia1, float soglia2) {

    if (ptrTree != NULL) {
        printTreeElemSoglie(ptrTree->left_ptr, soglia1, soglia2);
        if (ptrTree->right_ptr != NULL
            && (ptrTree->value >= soglia2)
            && (ptrTree->value <= soglia1)) {
            printf("%f \n", ptrTree->value);
        }
        printTreeElemSoglie(ptrTree->right_ptr, soglia1, soglia2);
    }
}
```

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2017/18
Prof. Michela Paolucci - 26/07/2018

- (14 punti)** Dato un array A di n elementi, scrivere:
 - Una funzione che ritorna il valore minimo e l'indice in cui è stato trovato
 - Una funzione che ripete n volte i seguenti passi: stampa il minimo, sovrascrive $+\infty$ in A nella posizione in cui è stato trovato il minimo
- (7 punti)** Date due liste A e B in forma collegata con puntatori, delle stesse dimensioni, scrivere la seguente funzione:
 - Che inserisce in una nuova lista i valori dispari di A e di seguito i valori pari di B
- (7 punti)** Dato un albero binario di ricerca, scrivere una sola funzione che:
 - Ritorna la somma dei valori dei nodi aventi almeno il figlio sinistro e stampa i valori dispari
- (2 punti)** Convertire da base 10 a base esadecimale il numero 746

Quesito 1a:

- Si scorre l'array e su aggiorna il minimo corrente.
- Ad ogni aggiornamento si salva anche la locazione in cui l'aggiornamento si è verificato.
- Si ritornano i due valori passando alla funzione gli indirizzi di due variabili.

```

void minArrayIndex(int * A, int n, int * min, int * minIndx){
    int i;
    int currentMin = +999999999;
    int currentMinIndx;

    for(i=0;i<n;i++){
        if (A[i] < currentMin){
            currentMin = A[i];
            currentMinIndx = i;
        }
    }

    *min = currentMin;
    *minIndx = currentMinIndx;
}

```

Quesito 1b:

- *Si scorre l'array e si invoca la funzione del punto 1a.*
- *Il valore di ritorno viene stampato*
- *La locazione in cui si è verificato il minimo viene sovrascritta*

```
void overwriteMinima(int * A, int n){
int i;
int min;
int minIndx;
for(i=0;i<n;i++){
    minArrayIndex(A,n,&min,&minIndx);
    printf("%i",min);
    A[minIndx] = +99999999;
}
}
```

Quesito 2:

- *Si visita la lista A e Per ogni nodo visitato: i) Se il valore è dispari, si inserisce in coda nella nuova Lista*
- *Si ripete lo stesso passaggio per la lista B*

```

void combinedList(struct list *ptrA,
                 struct list *ptrB,
                 struct list **ptrptrC) {
    init(ptrptrC);
    while (ptrA != NULL) {
        if(ptrA->value %2 !=0)
            suf_insert(ptrptrC, ptrA->value);
        ptrA = ptrA->next_ptr;
    }
    while (ptrB != NULL) {
        if (ptrB->value % 2 == 0)
            suf_insert(ptrptrC, ptrB->value);
        ptrB = ptrB->next_ptr;
    }
}

```

Quesito 3:

- *Si visita l'albero*
- *Per ogni nodo visitato: i) Se il nodo ha il figlio sinistro si aumenta il valore della variabile sumElem passata per puntatore; ii) Se il valore è dispari, si stampa*

```

void sumPrintTree(struct btree * ptrTree, int * sumElem) {
    if (ptrTree != NULL) {
        sumPrintTree(ptrTree->left_ptr, sumElem);
        sumPrintTree(ptrTree->right_ptr, sumElem);
        int val = ptrTree->value;
        if (ptrTree->left_ptr !=NULL ) {
            *sumElem = *sumElem + val;
        }
        if (val % 2 != 0)
            printf("Valore dispari: %d", val);
        }
    }
}

```

Quesito 4:

- *Si effettua la conversione da base dieci a base 2*
- *Si 'impaccano' i bit partendo dai 4 meno significativi e si convertono in base 16*
- *Si scrive il risultato finale*

$$[746]_{10} = [5 \cdot 10^2 + 4 \cdot 10 + 6]_{10} = [0111 \cdot 1010 \cdot 1010 + 0100 \cdot 1010 + 0110]_2 = \dots = [1011101010]_2$$

$$[0010]_2 = [2^1]_{10} = [2]_{10} = [2]_{16}$$

$$[1110]_2 = [2^3 + 2^2 + 2^1 + 0 \cdot 2^0]_{10} = [14]_{10} = [E]_{16}$$

$$[1010]_2 = [2^3 + 0 \cdot 2^2 + 2^1 + 0 \cdot 2^0]_{10} = [10]_{10} = [A]_{16}$$

$$[746]_{10} = [0111110001]_2 = [2EA]_{16}$$

Università di Firenze - Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2017/18
Prof. Michela Paolucci - 13/09/2018

1. (18 punti) Data una matrice A e uno scalare k, scrivere le seguenti funzioni:
 - a. Funzione che restituisce una nuova matrice $B = k*A$
 - b. Funzione che modifica la matrice originale A scrivendoci $A \leftarrow A/k$
 - c. Funzione che ritorna due nuove matrici
 $B=k*A$ e $C = A/k$
2. (7 punti) Data una lista in forma collegata con puntatori, scrivere:
 - a. Funzione che ritorna due liste: una con gli elementi divisibili per 3, l'altra con quelli divisibili per 5
3. (6 punti) Dato un albero binario di ricerca, scrivere una funzione che:
 - a. Stampa il valore contenuto nei nodi che hanno sia figlio destro che il figlio sinistro e restituisce la somma di tutti i nodi dell'albero
4. (2 punti) Effettuare le seguenti conversioni mostrando i passaggi:
 - a. $[10001110]_2 = [?]_{10}$
 - b. $[324]_{10} = [?]_2$

Università di Firenze - Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale
(A-L)

Prova Scritta di **Fondamenti di Informatica** A.A.
2017/18

Prof. Michela Paolucci - 10/01/2019

- (12 punti)** Date due matrici diagonali quadrate A e B di dimensione $n \times n$, scrivere le seguenti funzioni:
 - Funzione che calcola e restituisce il prodotto¹ fra le due matrici $C = A * B$
 - Funzione che modifica la matrice originale A scrivendoci il prodotto fra le due matrici $A \leftarrow A * B$
- (12 punti)** Date due liste A e B, scrivere:
 - Una funzione che restituisca la media della lista avente minima deviazione standard
 - Una funzione che prenda in ingresso una sola lista e ritorni una nuova lista con gli elementi diminuiti della media
- (6 punti)** Dato un albero binario di ricerca, scrivere una funzione che:
 - Stampa i valori dei nodi ordinati in modo crescente
 - Stampa i valori dei nodi ordinati in modo decrescente

Università di Firenze - Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A.
2017/18

Prof. Michela Paolucci - 14/02/2019

1. (12 punti) Dato un array di valori interi, scrivere:
 - a. Una funzione che ritorna un nuovo array con l'ordine di lettura degli elementi scambiato ("da sinistra a destra" in "da destra a sinistra". Es: 3456 -> 6543).
 - b. Una funzione che verifichi se un array letto al contrario rimane invariato (Es: 14541)
2. (12 punti) Date due liste A e B, scrivere:
 - a. Una funzione che elimini dalla lista B gli elementi della lista A.
 - b. Una funzione che restituisca una terza lista C contenente gli elementi ordinati sia di A che di B
3. (6 punti) Dato un albero binario di ricerca, scrivere una funzione che:
 - a. Trasferisce i valori in una lista in modo che la lista sia ordinata.
 - b. Trasferisce la lista del punto **a.** su un nuovo albero.

Università di Firenze - Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2017/18
Prof. Michela Paolucci - 18/04/2019

1. (12 **punti**) Date due matrici A e B, scrivere:
 - a. Una funzione che restituisca una nuova matrice $C = A + B^T$
 - b. Una funzione che restituisca una nuova matrice $A * B^T$ (prodotto righe per colonne)

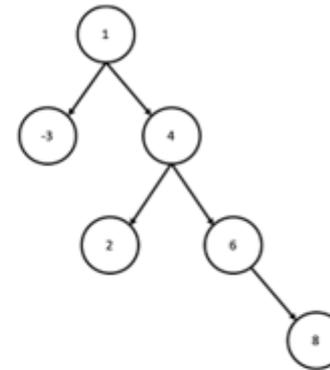
2. (10 **punti**) Data una lista di numeri interi in forma collegata con puntatori, scrivere:
 - a. Una funzione che controlli che la lista sia ordinata
 - b. Una funzione che accetta due liste e ritorna una nuova lista avente gli elementi delle due liste intercalati fra di loro. Esempio: da lista1=(2,5,9) e lista2= (4,1,8) si ottiene (2,4,5,1,9,8)

3. (8 **punti**) Dati due alberi binari di ricerca T_1 e T_2 a numeri interi, scrivere una funzione che genera un nuovo albero binario di ricerca che contenga gli elementi a comune tra T_1 e T_2

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2018/19
Prof. Michela Paolucci - 20/06/2019

1. (6 punti) Si consideri l'albero binario di ricerca in figura:

- Si scriva in uno dei possibili ordini di inserimento per ottenere lo stesso identico albero
- Dati gli stessi valori: $\{-3,1,2,4,6,8\}$ si determini un ordine di inserimento per il quale l'albero al termine degli inserimenti risulti bilanciato

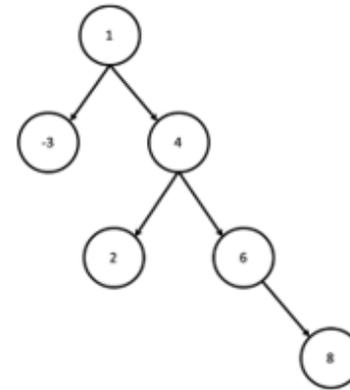


- (12 punti) Dato un albero binario di ricerca. Si scriva:
 - Una funzione che calcola la profondità del nodo più vicino alla radice contenente un valore dato e restituisca -1 in caso il nodo non sia presente.
- (12 punti) Date due liste A e B di lunghezza diversa, rispettivamente di lunghezza L_a e L_b , scrivere le seguenti funzioni:
 - Funzione che determina la lista di lunghezza minore e aggiunga in coda ad essa N elementi per arrivare alla lunghezza della seconda lista. I valori degli elementi aggiunti dovranno essere i primi numeri interi a partire da 5, ovvero: $\{5,6,7, 8, \dots\}$
 - Funzione che restituisca una terza lista C contenente il prodotto elemento per elemento dei valori delle due liste A e B

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2018/19
Prof. Michela Paolucci - 20/06/2018

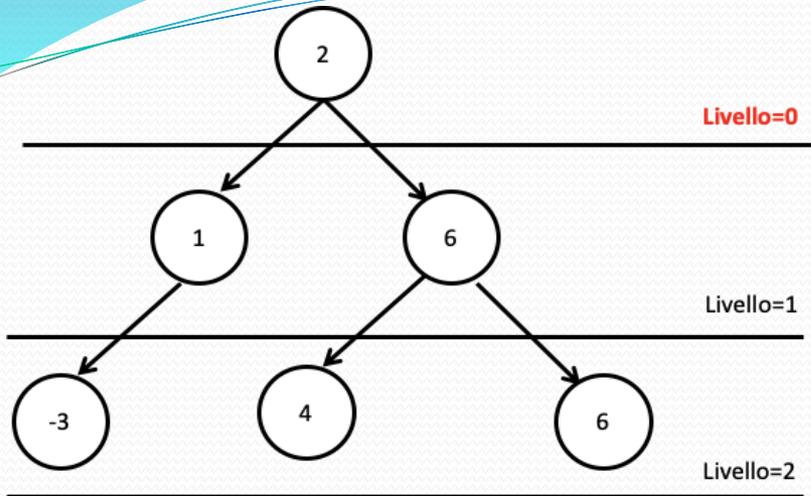
1. (6 punti) Si consideri l'albero binario di ricerca in figura:

- Si scriva in uno dei possibili ordini di inserimento per ottenere lo stesso identico albero
- Dati gli stessi valori: $\{-3,1,2,4,6,8\}$ si determini un ordine di inserimento per il quale l'albero al termine degli inserimenti risulti bilanciato



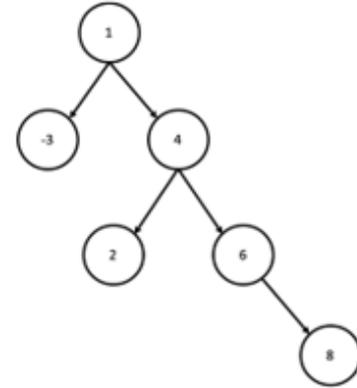
Esercizio 1a.

- Considerato che l'albero binario di ricerca in figura segue la convenzione *del valore sul figlio sinistro minore del figlio destro*. Ovvero detto *node* un nodo qualsiasi è sempre vero che $node \rightarrow left \rightarrow value < node \rightarrow right \rightarrow value$.
- Possiamo inferire un possibile ordine di inserimento:
 - $1, -3, 4, 2, 6, 8$
- oppure:
 - $1, 4, 6, 2, 8, -3$



1. (6 punti) Si consideri l'albero binario di ricerca in figura:

- Si scriva in uno dei possibili ordini di inserimento per ottenere lo stesso identico albero
- Dati gli stessi valori: $\{-3,1,2,4,6,8\}$ si determini un ordine di inserimento per il quale l'albero al termine degli inserimenti risulti bilanciato



Esercizio 1b.

- Un albero è **bilanciato** se ogni nodo in un livello diverso dall'ultimo o dal penultimo ha due figli.
- L'albero in figura non è bilanciato perché **-3** non ha nessun figlio ed è a livello 1, mentre l'albero ha massima profondità 3
- Esempio di inserimento: 2,1,-3,6,4,8
- Questo inserimento crea l'albero in figura. Come si può notare l'albero ha 3 livelli. Escludendo i nodi del penultimo (1) e ultimo livello (2) possiamo verificare che ogni nodo del livello 0 (ovvero la sola radice) ha due figli, di conseguenza l'albero è bilanciato.

Esercizio 2a.

- L'esercizio richiede di calcolare la profondità del primo nodo con il valore desiderato. Di conseguenza si deve adattare un algoritmo di ricerca per calcolare la profondità.
- Scegliendo la soluzione **ricorsiva** occorre passare una variabile *d* alla funzione con la profondità corrente incrementata di 1. Una volta trovato il valore ricercato si può restituire il valore *d*.

```
int compute_depth(struct btree* ptr, float value, int d){
    if (ptr){
        if (ptr->value==value) //if the value is founded, return current depth
            return d;
        else if (value < ptr->value){
            return compute_depth(ptr->left_ptr,value, d+1); //go left, and increment d
        }
        else
            return compute_depth(ptr->right_ptr,value, d+1);
    }
    else
        return -1; //we reached a null node so element was not found
}
```

Esercizio 2a. (alternativa)

- L'esercizio richiede di calcolare la profondità del primo nodo con il valore desiderato. Si deve quindi adattare un algoritmo di ricerca per calcolare la profondità.
- Scegliendo la soluzione iterativa si ha:

```
int compute_depth_it(struct btree* ptr, float value){
    int d=0;
    while(ptr!=NULL){
        if (ptr->value==value){
            return d;
        } else if (value < ptr->value){
            ptr = ptr->left_ptr;//go left,
            d++;//increment d
        } else {
            ptr = ptr->right_ptr;//go right,
            d++;//increment d
        }
    }
    return -1; //value not found
}
```

Esercizio 3a (1)

- Si crea una funzione `count_elements(...)` per determinare la lunghezza di una lista
- Si richiama tale funzione per confrontare la lunghezza delle due liste in esame e determinare quanti elementi aggiungere alla lista più corta in modo da avere due liste delle stesse dimensioni
- Gli elementi da aggiungere iniziano con un valore $i=5$, incrementato di 1 per ogni elemento successivo aggiunto alla lista più corta

//dichiarazione delle funzioni necessarie

```
int count_elements(struct list* ptr)
```

```
void add_int(struct list** ptrptrA, struct list** ptrptrB);
```

```
int count_elements(struct list* ptr) {
```

```
    int count = 0;
```

```
    while (ptr != NULL) {
```

```
        count++;
```

```
        ptr = ptr->next_ptr;
```

```
    }
```

```
    return count;
```

```
}
```

Esercizio 3a (2)

```
void add_int(struct list** ptrptrA, struct list** ptrptrB) {
    int countA = 0, countB = 0, diff = 0, i, value=5;
    countA = count_elements(*ptrptrA);
    countB = count_elements(*ptrptrB);
    if (countA == countB)
        return;
    else if (countB > countA) {
        diff = countB - countA; //numero di elem da
            //aggiungere alla lista più corta
        for (i = 0; i < diff; i++) {
            suf_insert(ptrptrA, value);
            value++;
        }
    }
    else { //countA > countB
        diff = countA - countB;
        for (i = 0; i < diff; i++) {
            suf_insert(ptrptrB, value);
            value++;
        }
    }
}
```

Esercizio 3b

- Si suppone che le liste abbiano la stessa dimensione
- La funzione che segue crea una nuova lista C t. c.:
 - $C[0] = A[0]*B[0]$
 - $C[1] = A[1]*B[1]$
 - ...

```
struct list* element_product(struct list* ptrA, struct list* ptrB) {  
    struct list *ptrC;  
    init(&ptrC);  
    while (ptrA != NULL) {  
        suf_insert(&ptrC, ptrA->value * ptrB->value);  
        ptrA = ptrA->next_ptr;  
        ptrB = ptrB->next_ptr;  
    }  
    return ptrC;  
}
```

Università di Firenze - Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2018/19
Prof. Michela Paolucci - 04/07/2018

1. **(12 punti)** Si consideri una immagine di dimensioni arbitrarie in scala di grigi rappresentata da una matrice di interi:

a. Si scriva una funzione che la ribalti orizzontalmente.



b. Si scriva una funzione che dato un riquadro definito con x, y , larghezza e altezza, crei una nuova matrice che rappresenti un riquadro dell'immagine



2. **(6 punti)** Si consideri una lista di interi ordinata. Si scriva una funzione che dati due valori m e M , modifichi la lista in modo che non contenga elementi x : $m < x < M$. Esempio: se la lista è $[1, 10, 12, 23, 34, 79]$ e $m=3$, $M=40$ la nuova lista sarà $[1, 79]$
3. **(12 punti)** Dati due alberi binari di ricerca a valori interi:
- Scrivere una funzione che determini i valori comuni e li inserisca in ordine decrescente in una lista
 - Scrivere una funzione che determini (ovvero restituisca) quale tra i due alberi contiene il nodo con il valore massimo e stampi tale valore

Esercizio 1a. (6 punti)

- Per ribaltare orizzontalmente l'immagine è necessario che:
$$\text{output}[i,j] = \text{input}[i, W-j],$$
 dove:
 - i , indica la riga della matrice e j la colonna;
 - W , indica la larghezza dell'array.
- Per semplicità usiamo x e y con: x indica colonna, y la riga.
- La matrice output è grande quanto la matrice input e viene allocata dalla funzione e restituita all'esterno come puntatore.

```
int* flip_image(int* input, int w, int h){
    //allocate output image
    int* output = malloc(sizeof(int)*w*h);
    int x=0, y=0;
    for (x=0; x < w; x++){
        for(y=0; y<h; y++){
            output[y*w + x]= input[y*w + (w-x)];
        }
    }
    return output;
}
```

Esercizio 1b

- In questo caso la matrice output ha dimensione diversa rispetto alla matrice iniziale, ed è specificata dall'utente con la dimensione del ritaglio
- Mantenendo la convenzione usuale per la rappresentazione di matrici, si fa scorrere:
 - i (riga di output) in $[0, h_crop]$
 - j (colonna di output) in $[0, w_crop]$
- Per leggere correttamente il riquadro occorre usare $(y+i)$ come indice di riga della matrice input e $(x+j)$ come indice di colonna.

```
int* crop_image(int* input, int x, int y, int w_crop,
               int h_crop, int w, int h){
    int *output=malloc(sizeof(int)*w_crop*h_crop);
    int i=0, j=0;
    for (i=0; i < h_crop; i++){
        for(j=0; j< w_crop; j++){
            output [i*w_crop+j] = input [(i+y)*w + j+x];
        }
    }
    return output;
}
```

Esercizio 2

- Sappiamo che la lista è ordinata per cui occorre scorrere i nodi fino a che non ci troviamo un nodo nell'intervallo di valori da cancellare.
- A quel punto si 'rimpiazza' l'attuale ptr con il successore finché non si esce dall'intervallo di valori.
- La soluzione funziona sia per liste ordinate in maniera crescente che decrescente.

```
void remove_interval(struct list** ptrptr, float min,
                    float max) {
    while(*ptrptr) { //iterate until the end
        //or until the list is empty
        if ((*ptrptr)->value < max &&
            (*ptrptr)->value > min) { //in range: remove
            struct list* ptr = *ptrptr; //save pointer
            (*ptrptr) = (*ptrptr)->next; //link to next elem.
            free(ptr); //free memory
        }
        else { //out of range: go on..
            ptrptr = &((*ptrptr)->next);
        }
    }
}
```

Esercizio 3a (1)

- Siano prtA e prtB i puntatori a due alberi binari
- La funzione void from_tree_to_list (...):
- Confronta i valori contenuti nei due alberi A e B, ovvero visita uno dei due alberi binari e per ogni nodo visitato, usa la funzione: Boolean search(struct btree * ptr, float target); per determinare se il valore del nodo è contenuto nel secondo albero
- Se tale valore è nel secondo albero, lo inserisce in una lista in ordine inverso

```
void from_tree_to_list(struct btree *ptrA,  
    struct btree *ptrB, struct list **ptrptr) {  
    if (ptrA != NULL) {  
        from_tree_to_list(ptrA->left_ptr, ptrB, ptrptr);  
        if(search(ptrB, ptrA->value))  
            ord_insert_inverse(ptrptr, ptrA->value);  
        from_tree_to_list(ptrA->right_ptr, ptrB, ptrptr);  
    }  
}
```

Esercizio 3a (2)

- Per inserire in ordine inverso basta modificare la funzione di inserimento ordinato
- Si scorre la lista finché il valore dell'elemento attualmente visitato è maggiore del valore da inserire, altrimenti si inserisce in testa

```
void ord_insert_inverse(struct list ** ptrptr, int value) {  
    /* inserimento in ordine inverso. Simile all'inserimento in  
    coda, salvo la condizione di inserimento */  
    while (*ptrptr != NULL && (*ptrptr)->value > value) {  
        ptrptr = &((*ptrptr)->next_ptr);  
    }  
    pre_insert(ptrptr, value);  
}
```

Esercizio 3b (1)

La funzione `struct btree * tree_with_max(...)`:

- Prende in ingresso i puntatori a due alberi binari di ricerca
- Determina il massimo di ogni albero (funzione `void max_bTree(...)`)
- Confronta i due massimi
- Stampa il massimo e restituisce il puntatore all'albero binario di ricerca che contiene il massimo

```
struct btree * tree_with_max(struct btree * ptrA, struct btree * ptrB){
    float maxA=0, maxB=0;
    max_bTree(ptrA, &maxA);
    max_bTree(ptrB, &maxB);
    if (maxA > maxB) {
        printf("Il massimo risulta: %f\n", maxA);
        return ptrA;
    }
    else {
        printf("Il massimo risulta: %f\n", maxB);
        return ptrB;
    }
}
```

Esercizio 3b (2)

La funzione `struct btree * tree_with_max(...)`:

- Prende in ingresso i puntatori a due alberi binari di ricerca
- Determina il massimo di ogni albero (funzione `void max_bTree(...)`)
- Confronta i due massimi
- Stampa il massimo e restituisce il puntatore all'albero binario di ricerca che contiene il massimo

```
struct btree * tree_with_max(struct btree * ptrA,  
* ptrB){  
    float maxA=0, maxB=0;  
    max_bTree(ptrA, &maxA);  
    max_bTree(ptrB, &maxB);  
    if (maxA > maxB) {  
        printf("IL massimo risulta: %f\n", maxA);  
        return ptrA;  
    }  
    else {  
        printf("IL massimo risulta: %f\n", maxB);  
        return ptrB;  
    }  
}
```

`struct btree`

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2018/19
Prof. Michela Paolucci - 25/07/2018

1. **(14 punti)** Si consideri una matrice di interi positivi di dimensione arbitraria
- Si scriva una funzione che crei una lista contenente tutti i valori > 0
Esempio (caso 2x6):

$$\begin{bmatrix} 12 & 0 & 0 & 0 & 0 & 14 \\ 0 & 22 & 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow [12, 22, 14]$$

- Si consideri una lista di lunghezza sconosciuta, contenente tutti gli elementi (diagonali) di una matrice diagonale. Si scriva una funzione che crei la suddetta matrice diagonale e la restituisca.
Esempio (caso 3x3):

$$[11, 2, 24] \rightarrow \begin{bmatrix} 11 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 24 \end{bmatrix}$$

2. **(2 punti)** Definire una struttura dati per rappresentare un database studenti da memorizzare come lista collegata con puntatori. Ogni studente deve essere rappresentato con **nome, cognome, matricola e media dei voti**.
3. **(14 punti)** Dato un albero binario di ricerca, scrivere:
- Una funzione che determini la profondità di un albero
 - Una funzione che, data una soglia, determini il numero di elementi appartenenti all'albero binario che stanno al di sotto stretto di tale soglia

Università di Firenze -- Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di **Fondamenti di Informatica** A.A. 2018/19
Prof. Michela Paolucci - 12/09/2019

1. **(12 punti)** Si consideri una matrice di interi:
 - a. Si scriva una funzione che restituisca un albero binario di ricerca contenente i soli elementi non nulli e multipli di 5
 - b. Si scriva una funzione che renda una matrice con n righe ed m colonne con gli elementi ordinati in base al seguente esempio
$$\begin{pmatrix} 3 & 10 & 5 \\ -4 & 4 & 9 \end{pmatrix} \rightarrow \begin{pmatrix} -4 & 3 & 4 \\ 5 & 9 & 10 \end{pmatrix}$$
2. **(8 punti)** Considerata una lista contenente un database studenti: nome, cognome, voto medio.
 - a. Scrivere una funzione di inserimento che rispetti l'ordine alfabetico del cognome.

NOTE: Si riportano le seguenti funzioni per la gestione delle stringhe:

- `int strcmp(const char* str1, const char* str2);` con:
 - **INPUT:**
 - **str1:** puntatore alla stringa 1
 - **str2:** puntatore alla stringa 2
 - **Output:**
 - 0, se le stringhe sono uguali
 - >0, se il valore ASCII del primo carattere non corrisposto della stringa str1 è maggiore del carattere della stringa str2
 - <0, se se il valore ASCII del primo carattere non corrisposto della stringa str2 è maggiore del carattere della stringa str1
- `char* strcpy(char* dest, const char* source);` con:
 - **INPUT:**
 - **dest:** puntatore all'array in cui viene copiata la stringa.
 - **src:** puntatore alla stringa da copiare
 - **Output:**
 - **puntatore a dest**

3. **(10 punti)** Dato un albero binario di ricerca a valori interi:
 - a. Scrivere una funzione che determini il numero di elementi pari totali e il numero di elementi pari il cui valore è al di sotto di una soglia prestabilita
 - b. Scrivere una funzione che restituisca il valore medio degli elementi

1. **(10 punti)** Si considerino **due liste** indipendenti della medesima lunghezza contenenti la sequenza dei voti e i CFU di ciascun esame.

Es.: voti: [27, 28, 30], CFU: [6,6,9], media pesata= 28,57

- a. Si scriva una funzione che restituisca la media pesata sui CFU della carriera. *Nell'esempio il primo esame è da 6 CFU ed ha avuto esito 27, l'ultimo è da 9 CFU con esito 30.*

$$\text{media pesata} = \frac{1}{\sum_{k=1}^{n\text{voti}} CFU_k} \sum_{i=1}^{n\text{voti}} \text{voto}_i \cdot CFU_i$$

- b. Si consideri infine il caso in cui si vogliono **escludere alcuni voti** dalla carriera fornendo alla funzione anche un'altra lista di Boolean il cui valore sarà TRUE per gli esami da includere nella la media e FALSE altrimenti.

Es.: voti: [27, 28, 30], CFU: [6,6,9], SEL: [TRUE, FALSE, TRUE], media pesata= 28,8

2. **(10 punti)** Si consideri un array di interi **N**, la cui lunghezza non è nota a priori ma solo al momento dell'esecuzione, che contiene unicamente i valori 1 e 0. **N** codifica **un intero positivo senza segno in base 2** in cui il primo bit è il più significativo.

- a. Si scriva una funzione che dato l'array restituisca l'intero positivo rappresentato dalla stringa di bit

Es.: N = [100] -> 4 (in questo caso $N[0] = 1$, $N[1]=0$, $N[2]=0$)

3. **(12 punti)** Si consideri una matrice di interi M, quadrata di dimensioni note solo al momento dell'esecuzione.

- a. **(5 punti)** Scrivere una funzione che determini se la matrice è simmetrica:

$$A_{ij} = A_{ji}. \text{ Es. di matrice simmetrica: } A: \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 6 \\ 3 & 6 & 8 \end{bmatrix}$$

- b. **(7 punti)** Scrivere una funzione che determini se la matrice contiene

almeno due righe uguali. Es. di matrice con righe identiche: $A: \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 6 \\ 1 & 2 & 3 \end{bmatrix}$

Università di Firenze -- Facoltà di Ingegneria

Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)

Prova Scritta di **Fondamenti di Informatica** A.A. 2018/19

Prof. Michela Paolucci - 23/01/2020 – 2° Appello INVERNALE

1. **(12 punti)** Si considerino **due liste** di lunghezza diversa. Creare una funzione per determinare:
 - a. Quale delle due liste ha il valor medio maggiore
 - b. Quale delle due liste ha lunghezza maggiore
 - c. Restituire una terza lista contenente tutti i valori di A che NON sono contenuti in B
 - d. Sovrascrivere A in modo che contenga: i valori di A e in seguito i valori di B

2. **(9 punti)** Dati due alberi binari di ricerca, creare una funzione per determinare:
 - a. Quale dei due alberi ha il valor medio maggiore
 - b. Quale dei due alberi ha il maggiore numero di elementi
 - c. Inserire nel primo albero anche gli elementi del secondo albero

3. **(9 punti)** Si consideri una matrice M che contenga l'entità delle spese giornaliere in euro previste per il mese di febbraio (righe = giorno del mese, colonne = tipo di spesa). Sapendo che il budget per tale mese è di 500 euro, scrivere una funzione capace di:
 - a. Determinare se la previsione di spesa rientra nel budget
 - b. Stampare la spesa media giornaliera
 - c. Restituire il risparmio mensile

Università di Firenze - Facoltà di Ingegneria
Corso di laurea in Ingegneria Meccanica e Gestionale (A-L)
Prova Scritta di Fondamenti di Informatica A.A. 2018/19
Prof. Michela Paolucci - 13/02/2020

- 1) **(8 punti)** Si considerino due liste non ordinate A e B di lunghezza diversa. Scrivere una funzione che:
- (2 punti)** Individui la lista più lunga.
 - (6 punti)** Sposti una parte dei nodi dalla lista più lunga alla più corta rendendole di eguale lunghezza (con al più un elemento di differenza).

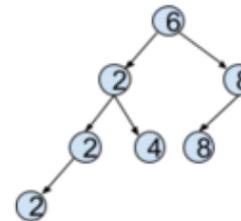
Esempio:

Prima della chiamata A: [21,2,4], B: [5,7,8,10,10,11]

Dopo la chiamata A: [21,2,4,5,7], B: [8,10,10,11]

- 2) **(12 punti)** Si consideri un albero binario di ricerca T. Scrivere una funzione che dato un valore v , ne conta le occorrenze n all'interno dell'albero.

Esempio: considerato T in figura: se $v=2$, $n=3$; se $v=10$ $n=0$; se $v=6$ $n=1$.



- 3) **(12 punti)** Data una matrice M di valori float la cui dimensione è specificata solo all'esecuzione del programma:
- (2 punti)** Scrivere una funzione restituisca una nuova matrice i cui valori negativi sono sostituiti da 0 e i valori positivi sono invariati.
 - (2 punti)** Scrivere una funzione che dato un indice di colonna restituisca un nuovo vettore contenente gli elementi della colonna.
 - (8 punti)** Scrivere una funzione che dato l'indice di una riga restituisce una matrice con una riga in meno.



Fondamenti di Informatica

Eng. Ph.D. Michela Paolucci

DISIT Lab <http://www.disit.dinfo.unifi.it/>

*Department of Information Engineering, DINFO
University of Florence*

Via S. Marta 3, 50139, Firenze, Italy

tel: +39-055-2758515, fax: +39-055-2758570

michela.paolucci@unifi.it



UNIVERSITÀ
DEGLI STUDI
FIRENZE

DINFO
DIPARTIMENTO DI
INGEGNERIA
DELL'INFORMAZIONE

DISIT
DISTRIBUTED SYSTEMS
AND INTERNET
TECHNOLOGIES LAB