



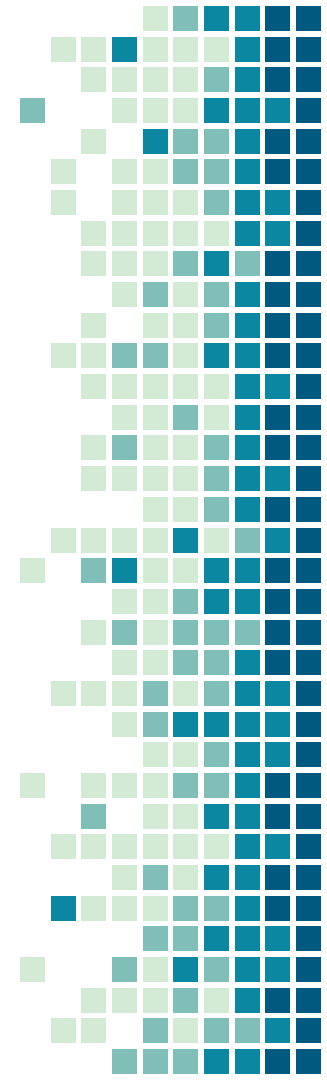
# CdL in Informatica - A.A. 2024 - 2025

## Programmazione 1 - Modulo 2

Lezione 4  
10/10/2024

Andrea Loddo

Federico Meloni - Alessandra Perniciano - Fabio Pili



# Argomenti

- Indentazione
- Generazione di numeri casuali
- Switch-case
- Operatore ternario



# Indentazione



# Indentazione

Che differenza c'è tra questi due codici?

```
#include <stdio.h>
int main(){int a=1;if(a==1){printf("a uguale a 1");}else{printf("a diverso da 1");}return 0;}
```

```
#include <stdio.h>
int main()
{
    int a=1;

    if(a==1)
    {
        printf("a uguale a 1");
    }
    else
    {
        printf("a diverso da 1");
    }

    return 0;
}
```

**Sintatticamente:** nessuna.

**Semanticamente:** nessuna.

Ma: quanto sono "**leggibili**" i codici?

# Indentazione: stili

- Regola GENERALE: istruzioni o blocchi secondari spostati di un tab.
- Automatico nella maggior parte degli IDE (Clion incluso...).
- **MOLTO** importante ai fini della valutazione d'esame.
- Leggibilità.

```
int main(){  
  
    int a=0, b=1;  
  
    if( a == 1 ){  
        if( b == 1 ){  
            printf("a e b veri");  
        }  
    } else {  
        printf("a falso");  
    }  
  
    return 0;  
}
```

```
int main()  
{  
    int a=0, b=1;  
    if(a == 1)  
    {  
        if(b == 1)  
        {  
            printf("a e b veri");  
        }  
    }  
    else  
    {  
        printf("a falso");  
    }  
  
    return 0;  
}
```

# Suggerimenti generali

1. Attenzione all'uso delle parentesi graffe nei blocchi quando è necessario che vengano eseguite più istruzioni.
2. Indentare il codice mandando avanti di un tab le istruzioni di ciascun blocco.
3. Rendere leggibile la condizione booleana all'interno delle parentesi ( ) dell'if, eventualmente disponendola su più righe o spezzando l' if in più istruzioni concatenate.
4. Attenzione all'operatore == (uguaglianza), tragicamente simile all'operatore di assegnamento ( = ).



# Generazione casuale

La funzione rand()



# Generazione pseudocasuale

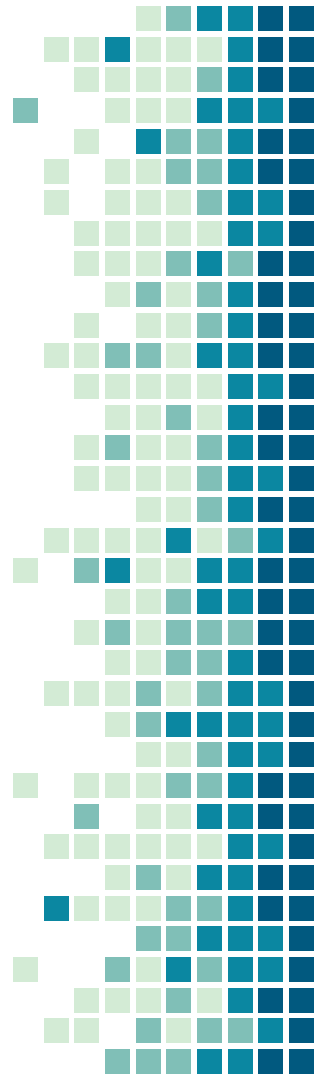
In C esiste un generatore di numeri, contenuto nella libreria **stdlib**.

```
#include <stdio.h>
#include <stdlib.h> // per la funzione rand()
int main()
{
    int numero;
    srand(50);      // inizializzazione del seme
    numero = rand();
    ...
}
```

La funzione `rand()` restituisce un numero compreso tra 0 e `RAND_MAX` (costante definita in `stdlib`, solitamente  $2^{15}$  o  $2^{31}$ )

La funzione `srand()` usa il parametro come **seme** per inizializzare il generatore di numeri casuali.

In realtà, usando lo stesso seme, si ottiene sempre la stessa sequenza numerica: generazione **pseudocasuale**.





# Generazione casuale

Per ottenere dei numeri veramente casuali, è necessario inizializzare alternativamente il seme, sfruttando la funzione `time()`.

```
#include <stdio.h>
#include <stdlib.h> // per la funzione rand()
#include <time.h>   // per la funzione time()
int main()
{
    int numero;
    srand(time(NULL)); // inizializzazione del seme
    numero = rand();
    ...
}
```

La funzione `time()` con parametro `NULL`, restituisce il tempo attuale in secondi trascorsi dal 01/01/1970.

Ora, eseguendo più volte il programma, otterremo un numero diverso.

# Pseudocasuale vs casuale: esempio

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

int main()
{
    int num;

    srand(0);          // generazione "pseudo"
    num = rand();
    printf("Numero generato: %d\n", num);

    srand(time(NULL)); // generazione casuale
    num = rand();
    printf("Numero generato: %d\n", num);
    return 0;
}
```

# Generazione casuale

**IMPORTANTE:** La funzione `srand()` deve essere utilizzata **UNA SOLA** volta all'interno del progetto! Meglio se nel `main`, **dopo** la dichiarazione delle variabili.

La funzione `rand()` restituisce numeri in un range molto ampio [0 – `RAND_MAX`]: tipicamente servono numeri casuali in range più piccoli.

Usiamo l'operatore `%` per restringere il range:

```
int num;

num = rand()%100;           // genera un numero compreso tra 0 e 99
num = rand()%100 + 50;      // genera un numero compreso tra 50 (offset) e 149

/* Per generare un qualsiasi numero random num in un dato intervallo
   [MIN, MAX] (estremi compresi): */
num = MIN + rand()%(MAX-MIN+1);
```

# Switch-case



# Switch-case

- Come verificare se una variabile assume un particolare valore in un insieme di valori predefiniti?
- Esempio. Supponiamo che:
  - la variabile **scelta** possa assumere valori interi tra 0 e N;
  - in corrispondenza di un dato valore, deve aver luogo l'esecuzione di un certo blocco di istruzioni.

```
int scelta;  
...  
  
if ( scelta == 0 )  
{  
    printf("Hai scelto 0");  
}  
else if ( scelta == 1 )  
{  
    printf("Hai scelto 1");  
}  
else if ( scelta == 2 )  
{  
    printf("Hai scelto 2");  
}  
...  

```

# Switch-case: sintassi

```
/* Sintassi dello switch-case in C */
switch (espressione)
{
    case costante1:
        /* blocco di istruzioni da svolgere se
        l'espressione è valutata costante1 */
        break;
    case costante2:
        /* blocco di istruzioni da svolgere se
        l'espressione è valutata costante2 */
        break;
    default:
        /* blocco di istruzioni da svolgere di
        default */
        break;
}
```

1. Viene valutata **espressione** e produrrà un risultato di tipo int (o char).
2. Se la valutazione è uguale a uno dei case dichiarati vengono eseguite le istruzioni (flusso di esecuzione) a partire dal case corrispondente.
3. Altrimenti, se non vi è alcuna corrispondenza, viene eseguito il blocco **default** (opzionale).
4. Il blocco del flusso di esecuzione deve essere esplicitato con l'istruzione **break**.

## Quando usarlo?

- Se la selezione dipende dal risultato di un'unica **espressione intera**
- Se i risultati dell'espressione sono **valori interi costanti noti a priori**

# Switch-case

```
/* Sintassi dello switch-case in C */
switch (espressione)
{
    case costante1:
        /* blocco di istruzioni da svolgere se
        l'espressione è valutata costante1 */
        break;
    case costante2:
        /* blocco di istruzioni da svolgere se
        l'espressione è valutata costante2 */
        break;
    default:
        /* blocco di istruzioni da svolgere di
        default */
        break;
}
```

```
/* Esempio di switch-case in C */
int x;
...
switch(x) {
    case 0:
        printf("Hai scelto 0");
        break;
    case 1:
        printf("Hai scelto 1");
        break;
    case 2:
        printf("Hai scelto 2");
        break;
    default:
        printf("Nessuna scelta");
}
```

Quando usarlo? Se la selezione:

- dipende dal risultato di un'unica espressione intera
- i risultati dell'espressione sono valori interi costanti noti a priori

# Switch-case: break o non break?

```
int numeroVocali = 0, numeroSpazi = 0, numeroConsonanti = 0;
char lettera = 'A';

switch (lettera)
{
    case 'A':
    case 'E':
    case 'I':
    case 'O':
    case 'U':
        numeroVocali++;
        break;
    case ' ':
        numeroSpazi++;
        break;
    default:
        numeroConsonanti++;
        break;
}
```



# Switch-case: considerazioni

- L'uso dello switch-case ha un senso quando è possibile effettuare confronti con **valori già noti**, e quando istruzioni da eseguire nei vari rami variano sensibilmente.
- Nel caso in cui sia necessario valutare più condizioni (ad esempio, se il valore di una variabile cade all'interno di una serie di intervalli) è necessario utilizzare il costrutto **if – else** in modo da coprire tutti i casi di interesse.

```
/* Esempio di codice sintatticamente scorretto
   che non può essere scritto in C */
switch (x)
{
    case (x < 0) :
        .....
        break;
    case ( (x >= 0) && (x <= 10) ) :
        .....
        break;
    ...
}
```

# Operatore ternario

- In C esiste un costrutto particolare, chiamato espressione condizionale o operatore ternario.
- Sintassi: **exp\_bool ? exp1 : exp2;**
- Viene valutata **exp\_bool**, se la sua valutazione è **true** viene eseguita **exp1** altrimenti **exp2**.

```
/* Esempio operatore ternario in C.  
Scopo: assegnare a k il maggiore tra i  
e j.  
i è maggiore di j? No, allora assegna j  
a k */  
  
int i=0, j=2, k;  
k = i > j ? i : j;
```

```
/* Esempio equivalente con if  
ed else. */  
int i=0, j=2, k;  
if( i > j) {  
    k = i;  
} else {  
    k = j;  
}
```

- Non aggiunge niente di nuovo ma offre un modo compatto per scrivere assegnamenti legati alla valutazione di un'espressione.

FINE!

Domande?

Autovalutazione



# Autovalutazione: generazione casuale

1. Quali sono le 3 operazioni base per poter generare pseudo-casualmente un numero?
2. Come posso generare un numero intero casuale tra 0 e 10?
3. Come posso generare un numero intero casuale tra -10 e 0?
4. Come posso generare un numero intero casuale tra -10 e 10?
5. Come posso generare un numero float casuale tra 0 e 1?
6. Come posso generare un numero dispari?
7. Come posso generare un numero pari?
8. Come posso generare un numero multiplo di 3?
9. Se genero due numeri casuali quante volte devo invocare la funzione `srand()`?



# Autovalutazione: switch-case

1. Il caso "default" è sempre obbligatorio all'interno di uno switch?
2. A cosa serve l'istruzione "break"?
3. Posso mettere all'interno di un case la valutazione di un'espressione non costante?
4. Che differenze ci sono tra switch e if? E quali sono le similitudini?



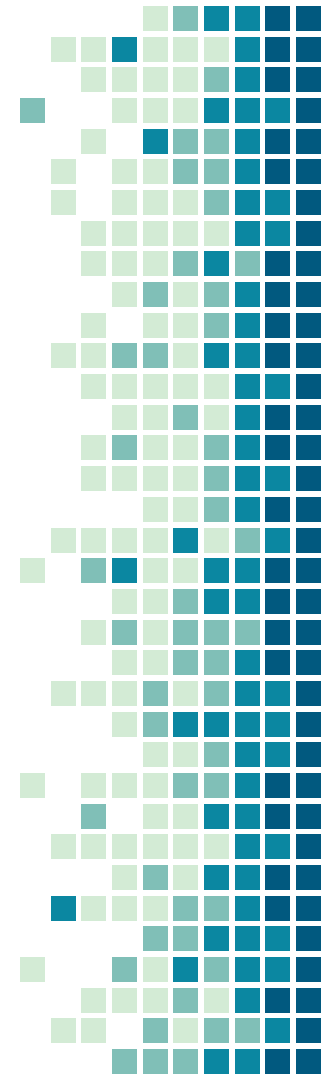
# Esercizi

Sulla generazione casuale



# Esercizio 1

- Scrivere un programma che chieda all'utente **min** e **max** di un intervallo.
- Generare, poi, 3 numeri casuali in quell'intervallo.
- Stampare, infine, i numeri generati.



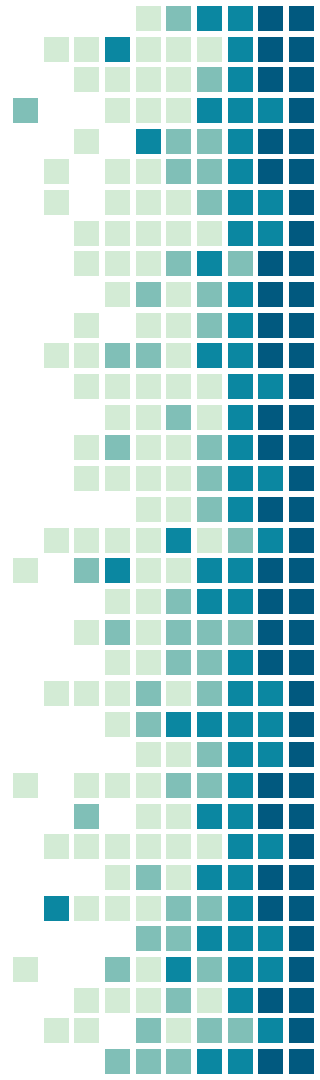


## Esercizi 2, 3, 4

2) Scrivere un programma che generi il peso di una persona da 3.00 a 120.00 kg (con due cifre decimali).

3) Scrivere un programma che, acquisito il peso in kg e l'altezza in metri, calcoli e stampi l'indice di massa corporea (BMI) secondo la formula  $BMI = \text{peso} / \text{altezza}^2$ .

4) Modificare il programma 3 sfruttando la generazione casuale utilizzata nel programma 2 e aggiungendo la generazione casuale dell'altezza usando come minimo 0.30m e come massimo 2.50m.



# Esercizio 5

Scrivere un programma che:

- Chieda all'utente di inserire due valori reali, che rappresentano il minimo e il massimo di un certo intervallo.
- Generi casualmente un certo valore casuale.
- Stampi a video se il valore generato è all'interno dell'intervallo oppure no.



# Esercizi

Sullo switch-case

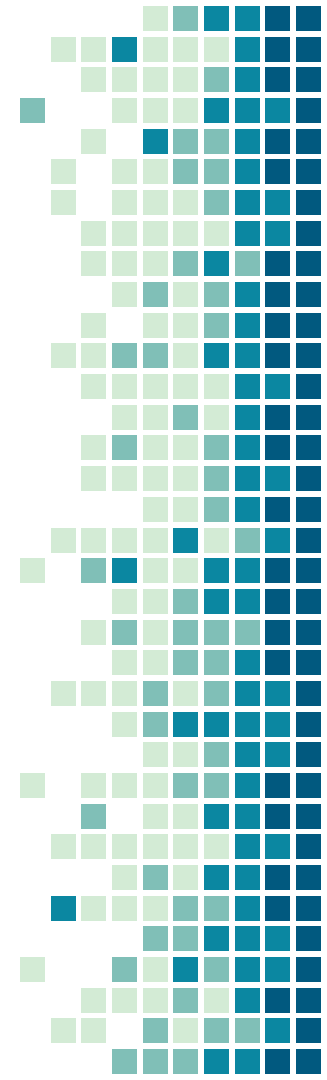


# Esercizio 0

Scrivere un programma che acquisisca in input un valore booleano, lo memorizzi in un'apposita variabile booleana e stampi a video se il suo valore è **true** o **false**.

Risolvere l'esercizio in 3 versioni distinte, usando:

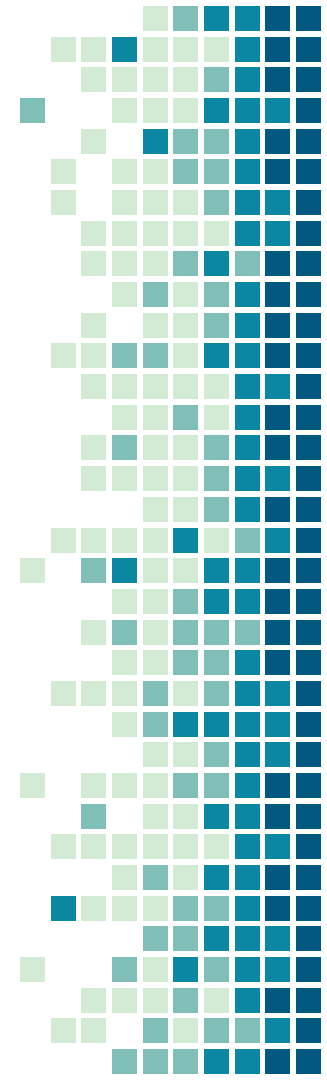
- If – else
- Switch – case
- Operatore ternario



# Esercizio 1

Scrivere un programma che, utilizzando lo switch-case, legga in input un voto rappresentato da un numero tra 2 e 10 e ne stampi il significato in questo modo:

- 10 → 'Ottimo'
- 9 → 'Distinto'
- 8 → 'Buono'
- 7 → 'Discreto'
- 6 → 'Sufficiente'
- 5 → 'Lievemente insufficiente'
- 4 → 'Gravemente insufficiente'
- 3 → 'Gravemente insufficiente'
- 2 → 'Gravemente insufficiente'



## Esercizio 2

- Scrivere un programma che generi un numero casuale tra 0 e 10 e chieda poi all'utente di inserire un valore tra 0 e 10.
- Se il numero inserito è maggiore di 10 o minore di 0 il programma deve stampare un messaggio di errore.
- Altrimenti, deve stampare "Numero uguale" se il valore inserito dall'utente è lo stesso generato casualmente o, in caso contrario, deve stampare la differenza in valore assoluto tra i due numeri.



# Esercizio 3

- Scrivere un programma che, inserito un valore numerico intero compreso tra 1 e 7 in input, stampi il corrispondente nome del giorno della settimana.
- Per esempio:
  - 1 → "Lunedì"
  - 6 → "Sabato"
- Realizzare, inoltre, un controllo sull'input utente e, se il valore non è tra quelli ammissibili, stampare un messaggio d'errore.



# Esercizio 4

Scrivere un programma per calcolare se un anno è bisestile oppure no. In particolare, il programma deve:

- Richiedere all'utente l'anno
- Verificare che l'anno sia maggiore di zero
- Nel caso in cui non lo sia, stampare un messaggio d'errore
- Diversamente, effettuare la verifica secondo il seguente schema: **un anno è bisestile se è divisibile per 400, oppure se è divisibile per 4 ma non per 100** ([Dettagli qui](#))
- In output, dire se l'anno è bisestile oppure no.





# Esercizio 5

- Scrivere un programma per la gestione dei biglietti del CTM.
- Si supponga che, rispetto alla tariffa piena, i pensionati usufruiscano di uno sconto del 10%, gli studenti del 15% e infine i disoccupati del 25%.
- Codificare lo stato dell'utente con una variabile char:
  - pensionati con una 'P',
  - gli studenti con una 'S' e
  - i disoccupati con una 'D'.
- Scrivere un programma che, richiesto il costo di un biglietto e l'eventuale condizione dell'utente, visualizzi l'importo da pagare.
- Suggerimento: potrebbe non servire l'if.



# Esercizio 6

- Scrivere un programma che chieda all'utente l'ammontare della spesa fatta in un negozio.
- In particolare, il negoziante effettua uno sconto del 5% per ogni spesa superiore a 120€ e del 10% per ogni spesa superiore a 350€.
- Visualizzare in output l'importo effettivo da pagare.
- Risolvere l'esercizio in due versioni:
  - con if-else
  - con soli switch-case



# Esercizio 7

Scrivere un programma che riceva in input un valore compreso tra 1 e 12, rappresentanti i mesi dell'anno, e stampi quanti giorni ha quel mese.

NB: Nel caso in cui il mese inserito sia febbraio, chiedere se l'anno è bisestile e gestire l'eventuale casistica in cui lo sia.

NB2: in caso di inserimento di un valore non ammissibile, comunicare all'utente un messaggio d'errore.

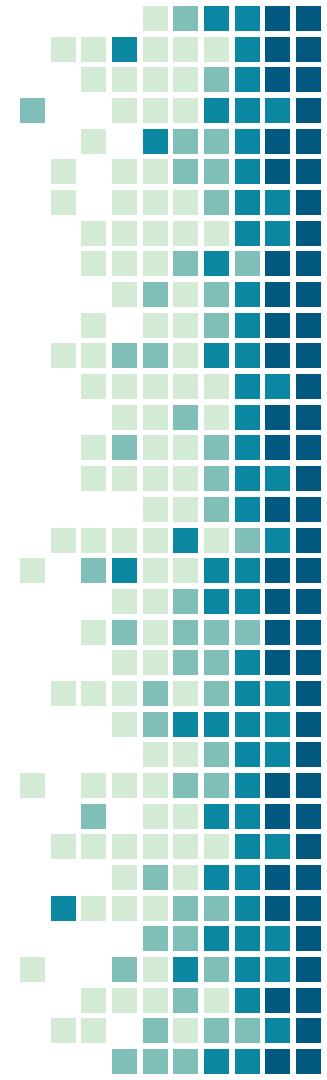


# Esercizio 8

Scrivere un programma che riceva in input un carattere e verifichi se il carattere inserito è una vocale oppure una consonante.

NB: gestire sia la casistica minuscola che maiuscola.

NB2: fornire il messaggio "carattere non consentito" nel caso in cui il carattere non sia una vocale.



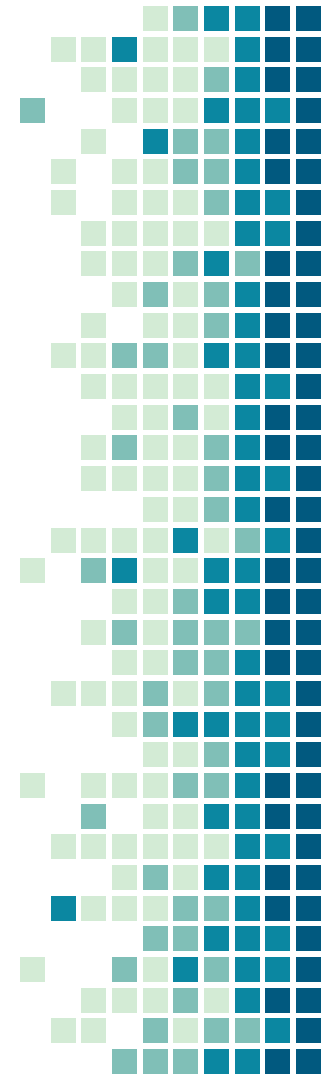
# Esercizio 9

Scrivere un programma che funzioni come calcolatrice, chiedendo all'utente l'operazione da svolgere ( +, -, \*, /, ^ ) e due operandi.

Per ^ (elevazione a potenza) utilizzare la libreria `math.h`.

Il programma deve svolgere l'operazione richiesta dall'utente e scrivere il risultato in output.

Usare lo `switch-case`.



# Esercizio 10

- Scrivere un programma che, acquisiti tre numeri, permetta di decidere all'utente se determinare il maggiore, determinare il minore, calcolare la somma o la media.
- Tale scelta dovrà avvenire inserendo uno tra i seguenti caratteri:
  - 'M' per il maggiore
  - 'm' per il minore
  - 's' per la somma,
  - 'a' per la media.

