

Università degli Studi di Ferrara

Corso di Laurea in Matematica - A.A. 2021 - 2022

Programmazione Lezione 10 – Esercizi in C

Docente: Michele Ferrari - michele.ferrari@unife.it

Nelle lezioni precedenti

- Abbiamo visto come gli elementi fondamentali della programmazione strutturata possono essere implementati in c
- Abbiamo visto come possiamo utilizzare variabili e costanti per elaborare i dati, come si dichiarano, alcuni tipi basilari e gli operatori fondamentali
- Abbiamo visto come visualizzare dati e messaggi sullo standard output (schermo) e come acquisire dati dallo standard input (tastiera)
- Abbiamo approfondito i tipi di dato
- Abbiamo visto come è possibile dichiarare e manipolare array di dati, anche multidimensionali
- Abbiamo trattato diversi problemi, tra cui ordinamento, ricerca del minimo, generare un numero casuale

In questa lezione

Proviamo a mettere assieme quanto fin qui appreso per risolvere qualche esercizio in C

Cosa ci serve?

- Un compilatore c (gcc)
- Un editor con la corretta syntax highlight

Nota: Con syntax highlighting o colorazione della sintassi si intende la caratteristica di un software, solitamente editor di testo, di visualizzare un testo con differenti colori e font in base a particolari regole sintattiche.

Consigli software: Windows

- Code:Blocks + mingw (editor+compilatore) (multiplatforma)
 - www.codeblocks.org/downloads/binaries
- Visual Studio Code (solo editor) (multiplatforma)
 - <https://code.visualstudio.com/>
- Gcc per windows (solo compilatore)
 - <http://www.mingw.org/> (attenzione al path!)

Consigli software: Mac OS

- Installare Command Line Developer Tools (compilatore gcc)
 - In un sistema connesso ad internet aprire il terminale e scrivere `gcc` seguito da invio: il sistema propone automaticamente l'installazione del software necessario
- Visual Studio Code (solo editor)
 - <https://code.visualstudio.com/>

Consigli software: Linux (Ubuntu)

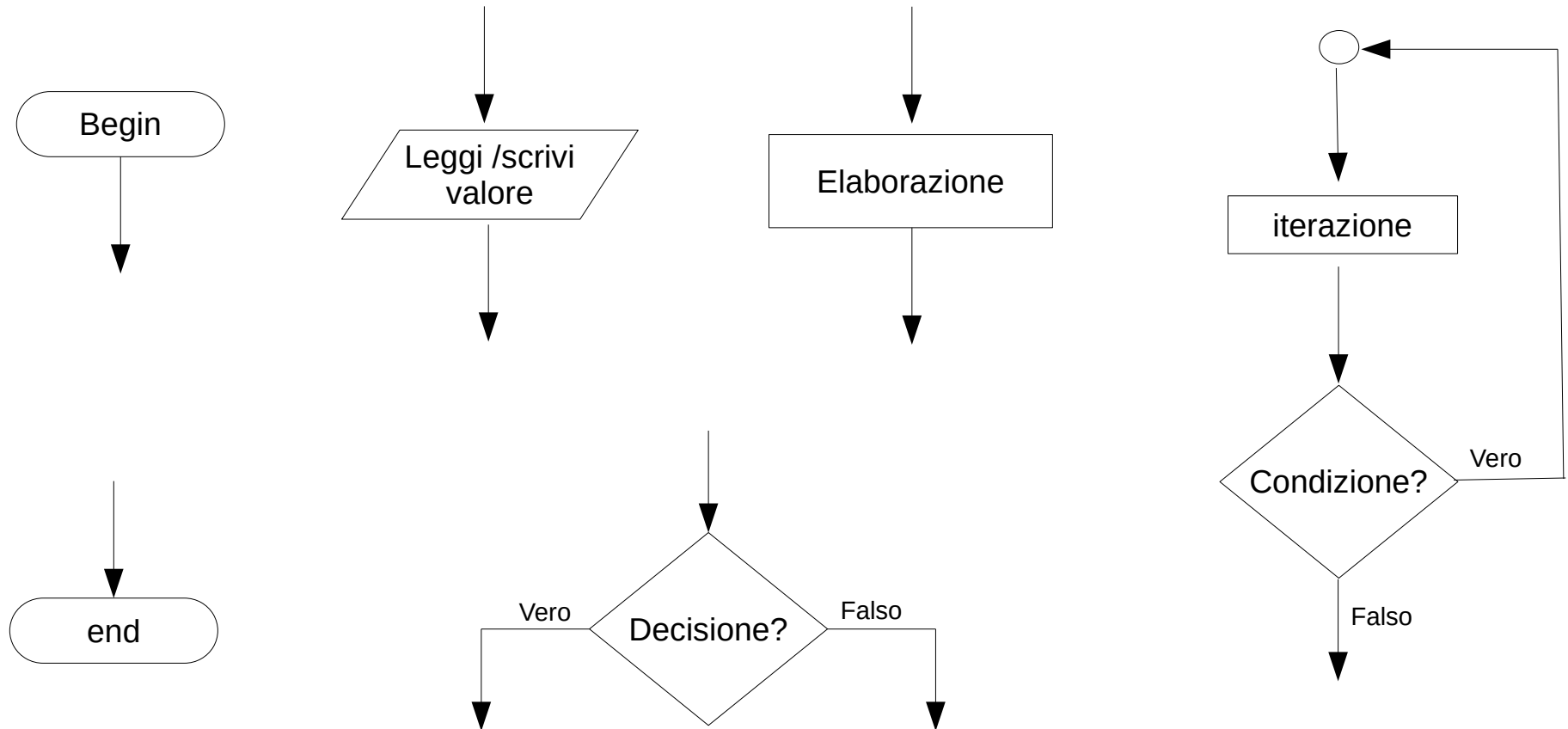
- Installare il pacchetto build-essential (per il compilatore gcc)
 - In un sistema connesso ad internet aprire il terminale e digitare:

```
sudo apt-get install build-essential
```

Inserire poi la password dell'utente (amministratore)

- Gedit – editor di testo (solo editor, preinstallato)
- Visual Studio Code (solo editor)
 - <https://code.visualstudio.com/>

Ricordiamo: Diagramma di flusso



Ricordiamo: I Tipi di dato in C

Tipo	Keyword
Intero	int
Reale	float
Reale a doppia precisione	double
Carattere	char
Indefinito	void

Ricordiamo: Sintassi C

Inclusione librerie

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

Funzione main

```
int main (void){  
    // istruzioni  
}
```

Assegnamento

```
a=b;
```

Operatori

```
a=b+c;
```

Costrutto if (selezione)

```
if (condizione){  
    // istruzioni  
}
```

Dichiarazione di una variabile

```
int a;
```

Iterazione

Ciclo for (controllo in testa, incremento in coda):

```
for(i=0;i<10;i++){  
    // istruzioni  
}
```

Ciclo while do (controllo in testa):

```
while (condizione){  
    // istruzioni  
}
```

Ciclo do while(controllo in coda):

```
do{  
    // istruzioni  
}while(condizione);
```

Dichiarazione di un array

```
int a[10];
```

Manipolare un array

```
int main(void)  
{  
    int numero[10];  
    int i, valore;  
  
    // inserimento  
    for (i=0; i<10; i++) {  
        scanf("%d", &valore);  
        numero[i] = valore;  
    }  
  
    // visualizzazione  
    for (i=0; i<10; i++) {  
        printf("%d", numero[i]);  
    }  
    return 0;  
}
```

Inizializzare un vettore grande N?

Premessa:

Iniziamo ad utilizzare diverse tecniche viste a lezione:

N è un parametro che può cambiare, potremmo utilizzare `#define` per indicare quanto grande farlo

Un vettore può essere inizializzato dall'utente con un ciclo e gli `scanf`, ma può essere anche inizializzato all'interno del codice, per mezzo di un ciclo e una qualche idea su come generare i valori o per mezzo della seguente sintassi per l'inizializzazione esplicita:

(esempio) `int array[N] = {1, 2, 3, 4, 5 ecc..};`

Esempio

```
#include <stdio.h>
#define N 5
int main (void)
{
    int array[N] = {1, 2, 3, 4, 5};
}
```

Esercizio 1

Scrivete un programma C opportunamente commentato che esegua le seguenti operazioni:

1. Inizializzi un array monodimensionale di interi
2. Esegue la rotazione dell'array

Esercizio 2

Scrivete un programma C opportunamente commentato che esegua le seguenti operazioni:

1. Dichiarare una matrice di interi $N \times M$
2. Inizializzare la matrice in modo che il valore di ogni elemento sia il risultato della somma dei propri indici
3. Visualizzare la matrice

Esercizio 3

Modificare il programma precedente in modo che i valori della matrice vengano generati in maniera casuale con un valore compreso fra 1 e 300.

Esercizio 4

Risolvere il seguente problema attraverso un programma in C opportunamente commentato:

In un concorso di intelligenza, N giudici esprimono un loro giudizio su M candidati, il giudizio è un valore numerico (compreso fra 0 e 5) inserito in una matrice di $N \times M$ elementi.

Si determini il candidato più intelligente e il giudice più severo.

Esercizio 5

Scrivere un programma che dichiari una matrice $N \times M$, ne inizializzi gli elementi con numeri casuali (vedi `rand()`) compresi fra 1 e 300 e ne esegua la visualizzazione degli elementi.

Ricavare in una seconda matrice, opportunamente dimensionata, la **trasposta** della prima matrice e visualizzarne gli elementi.

Esercizio 6

Scrivere un programma che dichiari una matrice, ne inizializzi gli elementi con numeri casuali (vedi `rand()`) compresi fra 1 e 300 e ne esegua la visualizzazione degli elementi.

Ricavare la **traccia** della matrice

Esercizio 7

Scrivere un programma in linguaggio C opportunamente commentato che:

1. Definisca una matrice $N \times M$ di valori interi i cui valori sono generati casualmente e la visualizzi
2. Richieda all'utente di inserire il valore di uno scalare
3. Moltiplichi la matrice per lo scalare e visualizzi il risultato

Esercizio 8

Scrivere un programma in linguaggio C opportunamente commentato che:

1. Definisca una matrice $N \times M$ di valori interi i cui valori sono generati casualmente e le visualizzi
2. Definisca una seconda matrice (di valori interi) opportunamente dimensionata i cui valori sono generati casualmente e le visualizzi
3. Calcoli il prodotto matrice per matrice, memorizzando i valori in una terza matrice opportunamente dimensionata e la visualizzi