

Università degli Studi di Ferrara

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

Programmazione Lezione 12 – 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, generazione di un numero casuale
- Abbiamo visto funzioni e funzioni ricorsive

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)

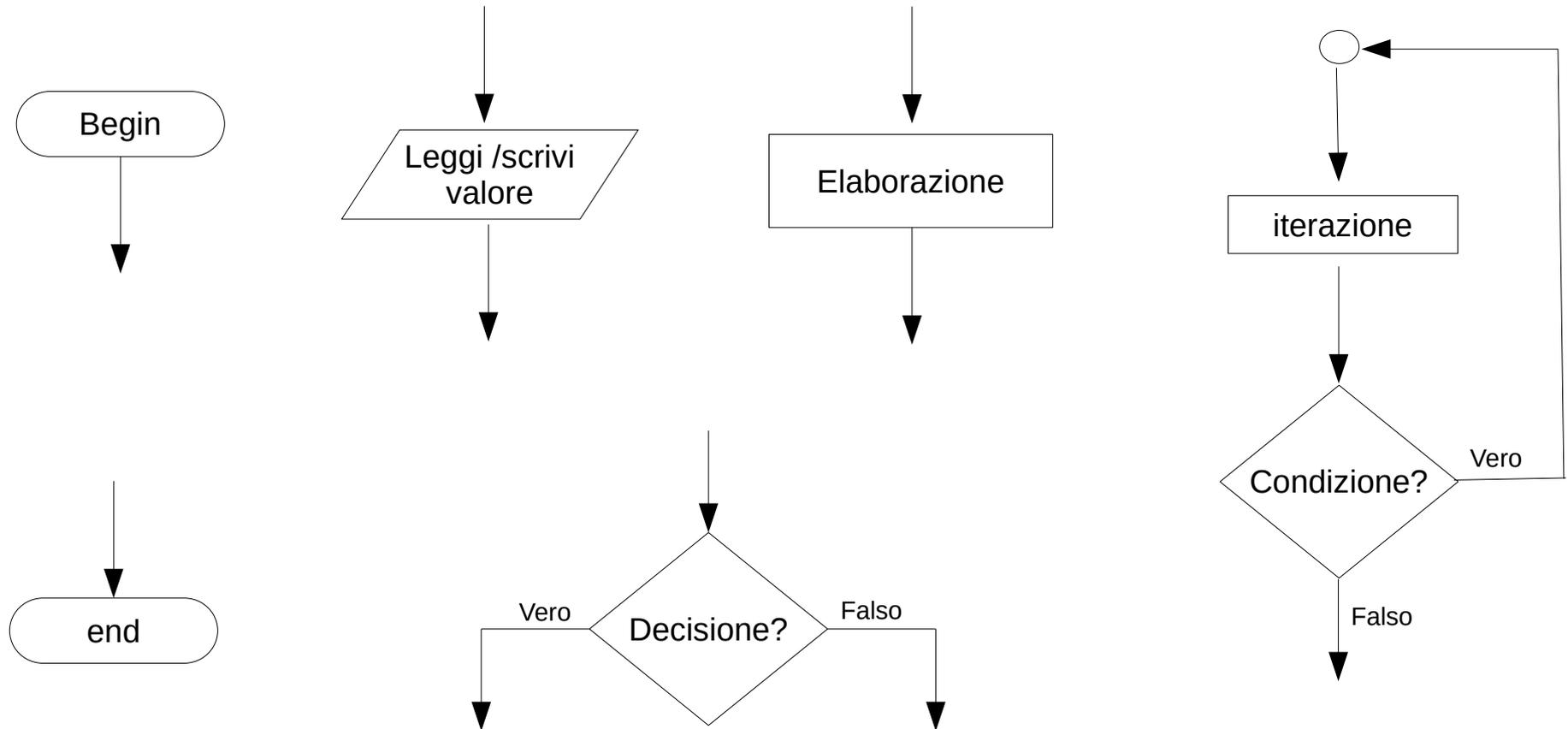
- Installare il pacchetto build-essential (per il compilatore gcc)
 - In un sistema connesso ad internet aprire il terminale e digitare:

```
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

Realizzare un programma in C opportunamente commentato che:

Definisca come funzione la moltiplicazione() fra due interi positivi, usando solo l'operatore di addizione e un ciclo for.

Definisca come funzione l'elevamento a potenza di un intero positivo per un altro, usando la funzione di moltiplicazione precedentemente definita.

Si testino le funzioni definite chiamandole dalla funzione main().

Esercizio 2

Realizzare un programma in C opportunamente commentato che:

Definisca `printfv()` come funzione il cui scopo è quello di visualizzare un vettore.

La funzione deve accettare 2 parametri, un array monodimensionale ed il numero di elementi che contiene l'array.

Per poter passare un array come parametro è necessario posporre l'operatore `*` (star).

Come prototipo della funzione, utilizzate:

```
void printfv(float *, int);
```

Esercizio 3

Realizzare un programma in C opportunamente commentato che calcoli la potenza di un numero n utilizzando una **funzione ricorsiva**.

Esercizio 4

Riguardate tutti gli esercizi svolti finora.

Individuate quali parti di codice sono ripetute e trasformatele in una funzione.

Riferimenti

Il corso di programmazione per il primo anno della Laurea Triennale in Matematica nasce con l'intento di unire ai principi di programmazione una conoscenza basilare di uno degli strumenti software più diffusi nell'ambito matematico: Matlab.

La prima parte del corso pertanto è una rielaborazione del programma di Programmazione per la Laurea Triennale in Informatica 15/16 (in particolare) e 16/17.

Parte del materiale originale da cui ho ricavato il percorso didattico, alcuni approfondimenti ed integrazioni possono essere trovati in:

- Lezioni di Programmazione 15/16 CdS Informatica - Giacomo Piva
- Lezioni di Programmazione 16/17 CdS Informatica - Marco Alberti