Richiami di Linguaggio C

Corso di C++

INFN - LNS

Corrado Santoro

Il Linguaggio C: struttura di un programma

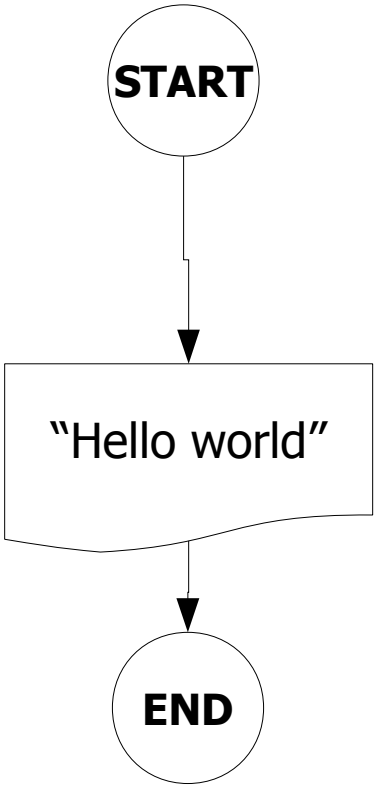
- Un programma C è un insieme di FUNZIONI che implementano l'algoritmo da realizzare
- Ogni funzione è composta da un insieme di comandi primitivi del linguaggio e da invocazioni (utilizzi) di altre funzioni esistenti nel programma stesso
- Dichiarazione di una funzione

```
NOME_FUNZIONE(ELENCO PARAMETRI)
{
   CORPO DELLA FUNZIONE
}
```

- Uso (invocazione di una funzione)

```
NOME_FUNZIONE(ELENCO PARAMETRI);
```

Verso il linguaggio C, il primo programma



hello.c

```
#include <stdio.h>

main()
{
  printf("Hello, world\n");
  getchar();
}
```

Dichiarare prima di usare

- Abbiamo visto che un programma C è un insieme di funzioni
- Ogni funzione va prima **dichiarata** e poi **usata**
- Lo stesso principio si applica alle **variabili**, le quali sono identificatori che possono assume un valore qualunque durante il corso del programma
- Le variabili vengono dichiarate specificando
  - Il **tipo**
  - Il **nome**
- La dichiarazione delle variabili è consentita **solo subito dopo** l'apertura di un blocco di istruzioni { ... }
- In C, il nome di una variabile, così come i nomi delle funzioni e le parole chiave, è **case-sensitive**

Due tipi del linguaggio C

- Intero (con segno):
  - Keyword: **`int`**
  - Range (32bit): [-2147483648, 2147483647]
- Intero senza segno
  - Keyword: **`unsigned int`**
  - Range (32 bit): [0, 4294967295]
- Reale (in virgola mobile):
  - Keyword: **`float`**
  - Range: [-10e38, +10e38]

Dichiariamo e usiamo le variabili

```
#include <stdio.h>

main()
{
   int i; /* dichiarazione di i, intera */
   float a, b; /* a e b sono due reali */
   ...
   i = 4;
   a = 10;
   b = (a + 1) * 3 / 15;
   ...
}
```

IMPORTANTISSIMO! La visibilità di una variabile (scope) è limitata al blocco in cui è stata definita.

Dichiarazione variabili: regola sintattica

- La dichiarazione di variabili in C è basata sulla seguente regola sintattica:
  - `TIPO NOME_VAR [ = CONST ], NOME_VAR [ = CONST] , .. ;`

- Esempi:
  - `float a = 10, b;`
  - `int i;`
  - `unsigned int j, k = 2;`

Stampiamo le variabili

- L'output di una variabile si ottiene con la funzione **printf**
- Per poter stampare qualunque variabile con printf, occorre specificare il formato di stampa
- In generale, **printf** è una funzione che prende un numero arbitrario di parametri:
  - Il primo parametro è una stringa che rappresenta il **formato** con cui il resto dei parametri va stampato

```
main()
{
   int i;
   i=10;
   printf("%d\n", i);
}
```

Stampiamo le variabili

- La stringa con il formato presenta sequenze del tipo "%carattere", che indicano
  - Il posto dove stampare la variabile
  - Il formato di stampa, stabilito dal carattere secondo la seguente tabella:
    - **%d → variabile intera con segno**
    - **%u → variabile intera senza segno**
    - **%f → variabile float**

```
main()
{
   int i;
   i=10;
   printf("Il valore di 'i' e' %d\n", i);
}
```

Esercizio

- **Scrivere un programma C per rappresentare, tramite tre variabili, le età di tre vostri amici**
- **Calcolare la media delle età**
- **Stampare:**
  - **Le tre età**
  - **La media calcolata**

Semantica delle operazioni matematiche

```
int enzo, paolo, luca;
float media;
media = (enzo + paolo + luca) / 3;
```

- Nonostante la variabile di destinazione sia float, tutti gli operandi delle operazioni sono interi → verrà eseguita una divisione **intera**

```
int enzo, paolo, luca;
float media;
media = (enzo + paolo + luca) / 3.0;
```

- Il divisore è una costante **forzata** in virgola mobile → verrà eseguita una divisione **in virgola mobile**