

Programmi C

- ✓ Struttura dei Programmi C
- ✓ Dichiarazione delle variabili
- ✓ Operazioni di Ingresso/Uscita

Struttura dei programmi C

Un programma C deve contenere, nell'ordine:

- una parte contenente le direttive per il compilatore;
- l'identificativo main
- due parti racchiuse da {}
 - una parte dichiarativa;
 - una parte esecutiva.

La parte dichiarativa contiene le seguenti sezioni:

- dichiarazione delle costanti;
- dichiarazione delle variabili.

Esempio: somma

Prof. Giuseppe Ascia

```
/* Direttiva per il compilatore*/
#include <stdio.h>

main()
{
    /* parte dichiarativa del programma */

    /* dichiarazione delle variabili */
    int a,b;
    int somma;

    /* dichiarazione delle costanti */
    const int c=2;
```

Fondamenti di Informatica

3

Esempio: somma

Prof. Giuseppe Ascia

```
/* parte esecutiva del programma */

printf("Inserire il primo numero");
scanf("%d",&a);
printf("Inserire il secondo numero");
scanf("%d",&b);
somma= c*(a+b);
printf("%d",somma);
}
```

Fondamenti di Informatica

4

Dichiarazione delle variabili

Prof. Giuseppe Ascia

Lo scopo della dichiarazione di variabili è di elencare le variabili utilizzate nella parte esecutiva e di attribuire a ciascuna di esse un tipo di dato.

Per tipo di dato si intende un insieme di valori e un insieme di operazioni che ad esse possono essere applicate.

La dichiarazione delle variabili assume la seguente forma:

specificatore_tipo elenco_identificatori_variabili;

```
int a;  
int b,c;  
float x,y;  
double z;  
char carattere,simbolo;
```

Una variabile per essere utilizzata in un programma C deve essere stata in precedenza dichiarata.

Fondamenti di Informatica

5

Tipi di dati C

Prof. Giuseppe Ascia

I tipi di dati che è possibile utilizzare in un programma C appartengono a due categorie:

- tipi di dati semplici
- tipi di dati strutturati.

Il linguaggio C prevede quattro tipi di dati semplici:

- **char** (carattere)
- **int** (intero)
- **float** (reale)
- **double** (reale in doppia precisione)

Fondamenti di Informatica

6

Il tipo int

Prof. Giuseppe Ascia

E' dato da un insieme di valori interi relativi e di operazioni su tali valori.

L'insieme di valori assunti dipende dal numero di bit usato per rappresentare i suoi valori.

Tale numero dipende dalla macchina su cui il programma deve essere eseguito.

Normalmente un **int** è rappresentato da 16 bit $\{-2^{15}, \dots, 2^{15}-1\}$

E' possibile fare in modo che un intero venga rappresentato con 32 bit mediante un **long int** (o **long**). $\{-2^{31}, \dots, 2^{31}-1\}$

Nel caso di **unsigned int** il numero intero rappresentato viene interpretato come numero non negativo appartenente all'insieme $\{0, \dots, 2^{16}-1\}$.

Nel caso di **unsigned long** i possibili valori appartengono all'insieme $\{0, \dots, 2^{32}-1\}$.

Fondamenti di Informatica

7

Il tipo int

Prof. Giuseppe Ascia

Le possibili operazioni applicabili al tipo int e suoi derivati sono:

=	<i>assegnamento</i>	==	<i>uguaglianza</i>
+	somma	!=	diversità
-	sottrazione	<	minore di
*	moltiplicazione	>	maggiore di
/	divisione	<=	minore o uguale
%	resto della divisione	>=	maggiore o uguale

Fondamenti di Informatica

8

Il tipi float e double

Prof. Giuseppe Ascia

Essi sono un'approssimazione dei numeri reali.

Lo spazio allocato per una variabile di tipo float e di tipo double dipende dalla particolare macchina.

Normalmente un **float** è memorizzato usando 4 byte, mentre un **double** è memorizzato usando 8 byte.

Le operazioni applicabili in C a dati di tipo float (double) sono le stesse di quelle per il tipo int.

Tipo char

Prof. Giuseppe Ascia

L'insieme di valori dei dati di tipo char è l'insieme di lettere, cifre, simboli disponibili sulla tastiera.

Oltre a questi caratteri esistono dei caratteri detti di controllo la cui scrittura comporta l'esecuzione di una operazione correlata alla visualizzazione dei dati.

Es. \n newline
 \t tabulazione orizzontale
 \b backspace

Le operazioni applicabili sono le stesse del tipo int.

Operazione di Ingresso/Uscita

Prof. Giuseppe Ascia

A rigori il linguaggio C non possiede istruzioni di ingresso/uscita, ma il programmatore può sfruttare alcune funzioni predefinite che realizzano tali operazioni. Per stampare si può usare la funzione

```
printf("stringa di controllo", elementi da stampare);
```

dove *stringa di controllo* è una stringa che viene stampata in uscita e che può contenere dei caratteri di formato preceduti dal simbolo %.

%d	stampa un	int
%ld	"	long (long int)
%f	"	float
%lf	"	double
%c	"	char

Per usare la funzione *printf* è necessario usare la direttiva

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

Fondamenti di Informatica

11

Operazione di Ingresso/Uscita

Prof. Giuseppe Ascia

Se **a** è una variabile di tipo **int** di valore 20,

```
printf("%d",a)
```

stampa sullo schermo il suo valore.

Se **a=10** e **b=20**, l'esecuzione di

```
printf("La somma tra %d e %d è %d", a,b,a+b);
```

stampa sullo schermo la seguente stringa:

La somma tra 10 e 20 è 30

Fondamenti di Informatica

12

Operazione di Ingresso/Uscita

Prof. Giuseppe Ascia

Per leggere si può usare la funzione

```
scanf("stringa controllo", elementi da leggere);
```

dove *stringa di controllo* è una stringa che può contenere dei caratteri di formato preceduti dal simbolo %.

Nell'elenco degli elementi da leggere compaiono i nomi delle variabili cui si desidera associare un valore.

I nomi delle variabili sono precedute dal simbolo &.

Per usare la funzione *scanf* è necessario usare la direttiva

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

Fondamenti di Informatica

13

Operazione di Ingresso/Uscita

Prof. Giuseppe Ascia

Es.

Data l'istruzione:

```
scanf ("%d %f", &a, &b) ;
```

se al momento dell'esecuzione l'utente inserisce i seguenti dati:

10 14.5

viene associato

alla variabile a il valore 10;

alla variabile b il valore 14.5;

Fondamenti di Informatica

14