



WWW.ALGORITMOSTEM.IT

SCIENCE TECHNOLOGY ENGINEERING MATHEMATICS

Appunti

Elementi di programmazione in linguaggio C

IIS2 - INFORMATICA

rev.0.9 - 02 set 2023

Draft version

Appunti in formato bozza, intesi esclusivamente di ausilio alle lezioni, che le integrano nelle descrizioni e nei ragionamenti su quanto viene riportato in queste pagine.

Licenza Creative Commons
CCBYNCND.

È consentita la condivisione del documento originale a condizione che non venga modificato né utilizzato a scopi commerciali, sempre attribuendo la paternità dell'opera all'autore

PROGRAMMAZIONE IN C

STRUTTURA DI UN PROGRAMMA in linguaggio C

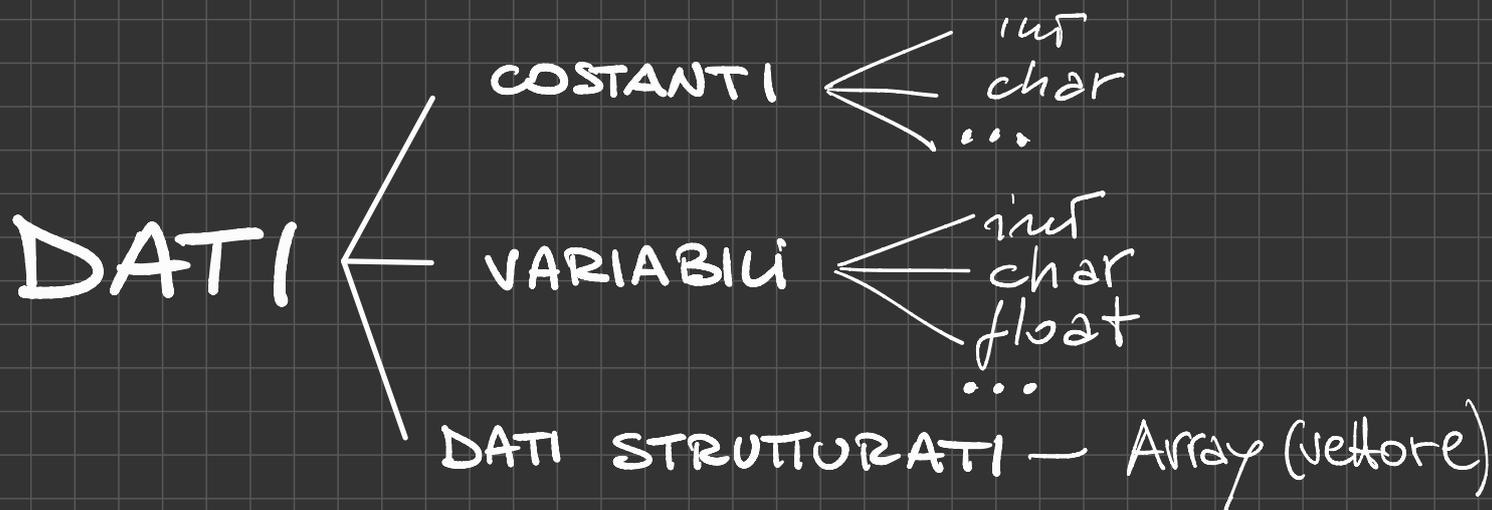
```
① #include <libreria>
② main( ) {
③   ISTRUZIONI;
}
```

- ① librerie di istruzioni -
Ad esempio, la libreria `stdio.h` include le funzioni di input/output
- ② Funzione principale che contiene tutte le istruzioni del programma
In generale, una funzione è definita:

tipo di dato in output parametri in ingresso istruzioni

tipo nome funzione (P₁, P₂, ... P_n) { }

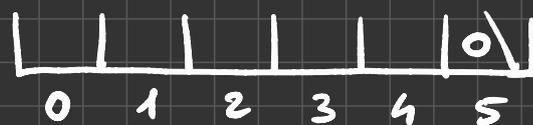
- ③ le istruzioni terminano con `;`
gruppi di istruzioni definiscono una funzione e sono racchiusi tra parentesi graffe



Successione indicizzata di elementi dello stesso tipo

$a[5]$

- int
- char
- float
- ...



esempio di dichiarazione `int v[5];`

esempi di dichiarazione con assegnazione

`int v[5] = { 10, 15, 1, 7, 9 };`

`char nome[4] = "nino";`

esempi di stampa

`printf("%d", a[0]);`

Stampa il primo elemento del vettore di numeri interi

`printf("%c", nome[1]);`

Stampa la seconda lettera del nome

`printf("%s", nome);`

Stampa lo stringa nome

esempio di inserimento

`scanf("%d", a[1]);`

inserisce il valore del secondo elemento del vettore a

DICHIARAZIONE DATI (variabili e costanti)

`int a`

`a` variabile di tipo intero

`int a=2`

`a` variabile di tipo intero con valore assegnato 2

`const int A=2`

`A` COSTANTE di tipo intero con valore assegnato 2

(in alternativa, si può dichiarare
#define A 2
prima della funzione main())

`int * p = &a`

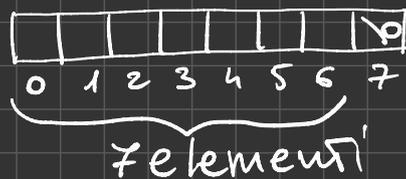
variabile puntatore all'indirizzo della variabile `a`

`char c="N"`

`c` variabile di tipo character con valore assegnato `N`

`char c[7]`

`c` variabile strutturata (ARRAY)
l'ultimo elemento è il "fine corso"



ISTRUZIONI DI ASSEGNAZIONE

$x = 23;$ la variabile x assume il valore 23

$w = "a";$ la variabile w assume il valore "a"

$y = z;$ il contenuto della variabile z viene memorizzato nella variabile y

$SUM = x + y$ la somma delle variabili x e y viene memorizzata nella variabile SUM

$t = (a + b) \cdot c / 2$ il risultato dell'espressione $(a + b) \cdot c / 2$ viene memorizzato nella variabile t

OPERATORI

Aritmetici $+$ $-$ $*$ $/$ $\%$ modulo (resto della divisione)
 $14 \% 6 = 2$

altri operatori nella libreria `math.h`,
ad esempio `pow` potenza
`sqrt` radice quadrata

Confronto $==$ uguale $!=$ diverso $>$ maggiore $>=$ maggiore uguale $<$ minore $<=$ minore uguale

Booleani AND OR NOT

ISTRUZIONI DI I/O

```
printf("Hello world");  
printf("il tuo nome è %s", nome);  
printf("la tua età è %d", ETA);  
printf("L'area del quadrato di lato %f è %f", LATO, AREA);  
scanf("%s", nome);  
scanf("%d", ETA);  
scanf("%c", UNCARATTERE);
```

ISTRUZIONI COMPOSITE

CONDIZIONALE

```
IF (condizione) { istruzioni }  
else { istruzioni }
```

esempio

```
⋮  
IF (x > y) { z = x }  
else { z = y }  
⋮
```

Iterativa

```
WHILE (condizione) { istruzioni }
```

esempio

```
⋮  
SOMMA = 0;  
SCANF ("%d", numero);  
WHILE (numero != 0)  
  { SOMMA = SOMMA + NUMERO; }  
printf ("%d", SOMMA);  
⋮
```

ESERCIZIO

Scrivere un programma in C che legge un vettore di interi di dimensione fissata e ne stampa la somma.

```
1 #include <stdio.h>
2 #define LUNG 5
3
4 int main() {
5
6 int vett[LUNG],i,sum=0;
7
8 printf("Inserire un vettore di interi di dimensione %d\n", LUNG);
9
10 for (i = 0; i < LUNG; i++)
11 {
12     printf("Inserire numero %d: ", i+1);
13     scanf("%d", &vett[i]);
14 }
15
16 /* Somma gli elementi del vettore */
17
18 for (i = 0; i < LUNG; i++) { sum=sum+vett[i];
19 }
20 /* Stampa la somma */
21 printf("\nSomma: %d ", sum);
22 return 0; }
```

con due
cicli FOR

```
1 #include <stdio.h>
2 #define LUNG 5
3
4 int main() {
5
6 int vett[LUNG],i,sum=0;
7
8 printf("Inserire un vettore di interi di dimensione %d\n", LUNG);
9
10 for (i = 0; i < LUNG; i++)
11 {
12     printf("Inserire numero %d: ", i+1);
13     scanf("%d", &vett[i]);
14     sum=sum+vett[i];
15 }
16
17 /* Stampa la somma */
18 printf("\nSomma: %d ", sum);
19 return 0; }
```

con un
ciclo FOR

```
1 #include <stdio.h>
2
3 #define LUNG 5
4
5 int main() {
6
7 int vett[LUNG], i, sum=0;
8
9 printf("Inserire un vettore di interi di dimensione %d\n", LUNG);
10
11 i=0; /* inizializzazione indice per condizione while */
12
13 while (i < LUNG) {
14
15     printf("Inserire numero %d: ", i+1);
16
17     scanf("%d", &vett[i]);
18
19     sum=sum+vett[i];
20
21     i=i+1;
22
23 }
24
25 printf("\nSomma: %d ", sum); /* Stampa la somma */
26 return 0; }
```

con istruzione
WHILE