C: tipi di dato semplici

Fondamenti di Programmazione

Tipi di dato

- Il **tipo di dato** specifica un insieme di *valori* e un insieme di possibili *operazioni* che possono essere applicate ad una variabile
- Ogni tipo ha una propria rappresentazione (codifica) in memoria attraverso un'opportuna sequenza di bit
- I linguaggi di programmazione permettono un certo livello di *astrazione* (possiamo ignorare come l'informazione viene rappresentata in memoria)
- Ogni variabile deve avere un tipo (assegnato dalla dichiarazione)

Classificazione dei tipi di dato

- **Tipi semplici**: per rappresentare informazioni semplici
- **Tipi strutturati**: per rappresentare informazioni costituite da diverse componenti

Tipi semplici

- **int** (interi)
- **float** (reali)
- **double** (reali con precisione doppia)
- **char** (caratteri)

Ogni tipo è associato ad un intervallo di valori che è legato alla quantità di memoria allocata (*dipende dalla macchina*)

Tipo int - interi

```
int n;
```

La dimensione di una variabile di tipo int (generalmente) è di 4 byte.

```
int n = sizeof(int);
```

signed o *unsigned*

short o *long*

`printf("%ld",...)` -> long

`printf("%hd",...)` -> short

Tipo float

```
float n;
```

Numeri con parte frazionaria: virgola mobile (*floating point*)

La dimensione di una variabile di tipo float (generalmente) è di 4 byte.

`printf("%f",...)` -> float
`printf("%.3f",...)` -> .000 float

`NB` : problema dell'approssimazione! 😕

Tipo double

Permette di avere una maggiore precisione

La dimensione di una variabile di tipo double (generalmente) è di 8 byte

Riduce l'effetto degli errori di approssimazione (arrotondamento)

Tipo char: caratteri

I caratteri possono assumere valori alfanumerici

La dimensione di una variabile di tipo char (generalmente) è di 1 byte.

```
char n;
n = 'A';
```

`printf("%c",...)` -> char

Altre funzioni standard: `getchar()` e `putchar()`

```
x = getchar();
putchar(x);
```