Un esempio di programma C

Il primo esempio - calcolo della media

Problema. Dati in input 3 numeri interi, calcolare e dare in output la loro media

Input: 3 numeri interi

Output: un numero reale.

Programma:

```
#include <iostream>
using namespace std;
int main()
    {int x,y,z;
    float m;
    cout << "dammi 3 numeri interi" << endl;
    cin >> x >> y >> z;
    m = (x + y + z) / 3.0;
    cout << "la media e' " << m;
    return 0;
    }
```

Commenti

- `#include <iostream>` (direttiva *include*): indica che il programma ha bisogno di utilizzare ("includere") delle funzioni predefinite per la gestione dell'input/output, le cui dichiarazioni sono contenute nel file `iostream`.
- `using namespace std;` (direttiva *using*): indica che il programma utilizza degli *identificatori* che appartengono allo "spazio dei nomi" standard (`std`); in particolare, i nomi `cin` e `cout` usati per identificare l'input e l'output del programma sono posti in questo spazio dei nomi.

- **main( )** indica che si tratta del *programma principale* (**int** e **main**: *parole chiave*; le parentesi tonde aperte e chiuse dopo **main** indicano assenza di parametri dall'esterno).
- Il programma principale ha di norma la seguente struttura:

```

main( )
  { DICHIARAZIONI
    +
    STATEMENT
  }

```

- La *dichiarazione di variabile*

```
int x, y, z
```

definisce tre nuove variabili, di nome *x*, *y* e *z*, tutte e tre di *tipo intero*.

La dichiarazione di variabile

```
float m;
```

definisce una nuova variabile di nome *m* e *tipo reale*.

Si noti che qualsiasi variabile prima di essere usata deve essere dichiarata in una parte del programma che precede l'uso della variabile stessa.

- Lo statement:

```
cout << "dammi 3 numeri interi" << endl;
```

indica un'operazione di output:

la stringa "dammi 3 numeri interi" viene inviata sul dispositivo di output standard (normalmente il monitor), seguita dall'invio del carattere speciale di "a capo".

L'esecuzione continua immediatamente.

Lo statement

```
cin >> x >> y >> z;
```

indica un'operazione di input:

si attende che tramite il dispositivo di input standard (solitamente la tastiera) vengano forniti al programma tre numeri interi, che verranno assegnati rispettivamente alle variabili x , y e z .

L'esecuzione continua soltanto quando tutti i valori attesi (tre costanti di intero in questo caso) sono stati forniti.

- Lo statement:

```
m = (x + y + z) / 3.0;
```

indica un'operazione di assegnamento:

l'espressione $(x + y + z) / 3.0$ viene valutata ed il suo risultato viene assegnato alla (ovvero, diventa il nuovo valore della) variabile m .

Se ad esempio, x vale 4, y vale 6 e z vale 3, il valore di m diventa 4.3333.

- Lo statement:

```
return 0;
```

indica che il programma termina restituendo all'*ambiente esterno* (= il Sistema Operativo da cui il programma viene chiamato) il valore 0 (= terminazione corretta).