Corso sul linguaggio C

Modulo 1

1 – Istruzioni semplici

Prerequisiti

- Saper progettare algoritmi
- Saper descrivere algoritmi (NLS, ddf)
- Conoscere il concetto di espressione

Introduzione

In questa lezione si vedranno alcuni concetti fondamentali della programmazione in C, utili per scrivere i primi programmmi.

Nella prossima Unità si vedrà come provare al computer i programmi realizzati.

Informazioni generali

Per poter programmare occorre conoscere:

- l'algoritmo risolutivo del problema
- la struttura del programma
- gli elementi del programma

In carattere **grassetto** indichiamo le parole chiave del linguaggio, mentre in carattere *corsivo* si indicano elementi che vengono specificati solo in fase di codifica.

- Il C è *case sentitive* ossia distingue tra maiuscolo e minuscolo, quindi bisogna fare attenzione durante la scrittura delle istruzioni.

Concetti fondamentali

Un' **istruzione** è una frase scritta in un determinato linguaggio di programmazione che indica alla macchina un singolo compito da svolgere.

Questo compito sarà eseguito quando tutta la sequenza di istruzioni sarà conosciuta dalla macchina.

La sequenza di istruzioni costituisce il **programma**.

Eseguire un programma significa svolgere tutte le sue istruzioni in sequenza, partendo dalla prima.

Concetti fondamentali

I dati su cui il programma opera sono contenuti in:

- variabili
- costanti
- espressioni

Il dato contenuto in una variabile *può essere modificato* al contrario di quello contenuto in una costante.

Variabili

La dichiarazione di una **variabile** segue la sintassi:

tipo ident;

dove

- *tipo* indica il tipo della variabile, che può essere:
  - **int** – interi
  - **char** – caratteri
  - **float** – numeri reali in semplice precisione
  - **double** – numeri in doppia precisione
- *ident* è il nome (identificatore) che il programmatore ha assegnato alla variabile
- Esempi:
  - **int** somma**;** **/*** Dichiara la variabile di nome *somma* di tipo **int** ***/**
  - **int** var1**,** var2**;** **/*** Dichiara due variabili, ognuna di tipo **int** ***/**

Le frasi racchiuse tra /* … */ si dicono **commenti** e sono descritti più avanti

Costanti

La sintassi per dichiarare le costanti è:

#define *ident espressione*

dove

- *ident* è il nome della costante
- *espressione* può essere un intero, un reale o una **stringa costante**,
  - **#define** MESSAGGIO "Dato inserito"

Osservazioni:

- Con il termine **stringa** intendiamo una sequenza di caratteri
- Per convenzione le *costanti si indicano con nomi in caratteri MAIUSCOLI*

Espressioni

All'interno delle istruzioni possono figurare **espressioni**

Una espressione può essere:

- una costante
- una variabile
- una combinazione di espressioni mediante **operatori**

Esempi:

- 5 è un'**espressione costante intera**
- 3+6 è un'**espressione numerica intera**
- X+5 è un'**espressione variabile**
- 'a' è un'**espressione costante carattere.**

Successivamente si vedranno in dettaglio i vari tipi di operatori

Struttura di un programma in C

```
#include <stdio.h>

int main()
{
   dichiarazioni variabili
   blocco istruzioni
   return(0);
}
```

Le frasi in *corsivo* indicano un elemento generico che sarà specificato successivamente nella fase di codifica del programma

Questo è il *blocco delle dichiarazioni*. In questa parte si scrivono alcuni elementi che serviranno in seguito al programma.

In questa parte si scrivono in sequenza, le istruzioni che dovranno essere eseguite.

Istruzioni semplici

Iniziamo a considerare le **istruzioni semplici:**

- lettura
- stampa
- calcolo e assegnazione

Un programma formato solo da questi 3 tipi di istruzioni *viene sempre eseguito in sequenza* dall'inizio alla fine.

Lettura

Lettura

scanf (*formato*, *ident*);

dove

- *formato* è una stringa che specifica il tipo del dato della variabile *ident* ("%d" per gli interi, "%c" per i caratteri, "%s" per le stringhe, come vedremo tra breve)

L'istruzione di lettura consente a chi usa il programma di immettere in esso valori mediante la tastiera.

Vedremo in seguito il significato del carattere **&** (detto *ampersand*)

Esempi:

- **scanf** ("%d", **&**numero); /* legge la variabile intera *numero* */
- **scanf** ("%c", **&**ch); /* legge il carattere *ch* */

Stampa

Stampa

printf (*formato*, *ident*);

- dove *formato* e *ident* hanno il significato illustrato per la **scanf**().

L'istruzione di stampa consente di emettere a video il valore che in quel momento possiede una qualunque variabile.

Esempi:

printf ("%c", ch); /* stampa il contenuto di *ch* */

Stampa

Si possono ottenere in stampa effetti speciali mediante le **sequenze di escape** da abbinare all'istruzione **printf()**. Alcune sono indicate nella tabella.

Carattere di escape	Effetto
\n	A capo
\a	Segnalazione acustica
\t	Tabulazione orizzontale
\b	Una battuta indietro
\r	Ritorno a capo sulla stessa riga
\0	Carattere di fine stringa
\f	Salto pagine
\ddd	Ottale (d è una cifra ottale)
\xdd	Esadecimale (d è una cifra esadeimale)
\\	Stampa il carattere '\'

Stampa

Esempi di sequenze di escape

printf ("Emetto un beep (\\a) - premere un tasto \a“);
printf ("Emetto un carattere di carriage return (\\r) - premere un tasto\r“);
printf ("Questo e' l'uso del tabulatore (\\t)\n“);
printf ("Oggi\te'\til\t19\tfebbraio\t1993 - premere un tasto“);
printf ("Questo e' il carattere \101 di codice ottale \\101 - premere un tasto“);
printf (“Questo e' il carattere \x41 avente codice hex \\x41\n“);

printf (“Elenco valori\n”);
printf (“Nome\tTel.”);

Stampa la frase “Elenco valori” e va a capo

Stampa le stringhe separate dal carattere di tabulazione

Lettura e stampa formattate

Il *formato* previsto nelle istruzioni **scanf** () e **printf** () serve al programmatore per sceglier il tipo dei dati. Per questo motivo si dicono **lettura e stampa formattate**.

Il *formato* rappresenta un *carattere di conversione*.

Esempi:

scanf (“%d”, &valore); /* legge variabile intera */
printf (“%f”,area); /* stampa variabile float */
printf (“%s %d”, cognome, totale); /* stampa un intero ed una stringa */
printf (“Il quadrato di %d e’ %d”,num, quad_num);

Lettura e stampa formattate

I più comuni *caratteri di conversione* sono quelli riportati in tabella:

Formato	**Converte in**
%d	Intero
%f	Reale
%e %E	Reale in notazione scientifica
%c	Reale in doppia precisione
%s	Stringa
%x	Esadecimale
%p	Indirizzo della variabile
%o	Ottale

Lettura e stampa formattate

In particolare per i caratteri di conversione "%d" e "%f" esiste anche la possibilità di indicare la precisione:

Esempi:

printf ("%3d", somma); /* stampa su 3 posizioni */

printf ("%5.2f", misura); /* stampa di un numero reale su 5 posti con 2 decimali */

Assegnazione

Assegnazione e calcolo

ident = *espressione*;

- dove *espressione* può essere una costante, una variabile o una qualunque espressione che viene assegnata alla variabile *ident.*

Esempi:
- X = 1;
- Nome = “Rossi”;
- S = x + y;

Le direttive

```
#include <stdio.h>
#define RISULTATO "La somma e': %d\n"
int main()
{
   int a,b,c;
   printf("Primo valore: “);
   scanf(“%d”, &a);
   printf("Secondo valore: “);
   scanf(“%d”, &b);
   c=a+b;
   printf (RISULTATO,  c);
   getchar();
   return 0;
}
```

Questo è un esempio di **direttiva**. Le direttive iniziano con il simbolo “#”.

In questo caso la direttiva serve per includere una **libreria**.

Le librerie sono particolari file che contengono l’espansione di alcune istruzioni in modo che il programmatore possa limitarsi a fare solo riferimento ad esse.

Nel nostro caso la libreria **stdio.h** consente di usare le funzioni **scanf**() e **printf**()

Uso di costanti

```
#include <stdio.h>
#define RISULTATO "La somma e': %d\n"
int main()
{
   int a,b,c;
   printf("Primo valore: “);
   scanf(“%d”, &a);
   printf("Secondo valore: “);
   scanf(“%d”, &b);
   c=a+b;
   printf (RISULTATO,  c);
   getchar();
   return 0;
}
```

Quest’altra direttiva definisce le **costante** RISULTATO.

Una costante è una particolare variabile che assume un valore all’inizio del programma e che non varierà più durante tutta l’esecuzione.

La costante in questo caso ha nome RISULTATO e contiene la stringa "La somma e': %d\n”

Uso di variabili

```
#include <stdio.h>
#define RISULTATO "La somma e': %d\n"
int main()
{
   int a,b,c;
   printf("Primo valore: “);
   scanf(“%d”, &a);
   printf("Secondo valore: “);
   scanf(“%d”, &b);
   c=a+b;
   printf (RISULTATO,  c);
   getchar();
   return 0;
}
```

In questa parte vengono dichiarate le **variabili** del programma. In questo caso si usano 3 variabili: *a,b* e *c*.

Queste variabili sono destinate ad ospitare valori interi essendo precedute dal simbolo **int**.

Uso dell'assegnazione

```
#include <stdio.h>
#define RISULTATO "La somma e': %d\n"
int main()
{
   int a,b,c;
   printf("Primo valore: “);
   scanf(“%d”, &a);
   printf("Secondo valore: “);
   scanf(“%d”, &b);
   c=a+b;
   printf (RISULTATO,  c);
   getchar();
   return 0;
}
```

Con questa istruzione si **assegna** a c il valore della somma tra a e b.

espressione può essere una qualunque espressione in senso aritmetico (tra numeri), algebrico (con variabili) o stringa (tra stringhe)

I commenti

```
#include <stdio.h>
#define RISULTATO "La somma e %d\n ':"
int main()
{
   int a,b,c;   /* dichiarazione variabili */
   printf("Primo valore: “);
   scanf(“%d”, &a);  /* lettura di a */
   printf("Secondo valore: “);
   scanf(“%d”, &b);  /* lettura di b */
   c=a+b;
   printf(RISULTATO,  c);
   getchar();
   return 0;
}
```

Questi sono esempi di **commenti**. Un commento rappresenta note che il programmatore scrive all'interno del programma e che non saranno elaborate durante l'esecuzione.

La sintassi per i commenti prevede:

Il simbolo "/*" per **inizio commento**

Il simbolo "*/" per **fine commento**

Un commento può occupare anche più di una riga.

Terminatore istruzioni

```
#include <stdio.h>
#define RISULTATO "La somma e %d\n ':"
int main()
{
   int a,b,c;   /* dichiarazione variabili */
   printf("Primo valore: “);
   scanf(“%d”, &a);  /* lettura di a */
   printf("Secondo valore: “);
   scanf(“%d”, &b);  /* lettura di b */
   c=a+b;
   printf(RISULTATO,  c);
   getchar();
   return 0;
}
```

Si noti che ogni istruzione termina con un simbolo di “;”. Questa è una precisa regola sintattica del linguaggio.

Argomenti

- Concetti fondamentali
- Variabili
- Costanti
- Espressioni
- Struttura di un programma in C
- Istruzioni semplici
- Lettura
- Stampa
- Lettura e stampa formattate
- Assegnazione
- Le direttive
- Uso di costanti
- Uso di variabili
- Uso dell’assegnazione
- I commenti
- Terminatore istruzioni

Attività

- Scrivere un programma che stampi le frasi
  Prima linea
  Seconda linea
  Terza linea
- Scrivere un programma che leggendo da input le coordinate intere di due punti del piano, calcoli le coordinate del punto medio del segmento che li unisce.

Altre fonti di informazione

- J. Purdum, C - ed. Jackson
- Romagnoli Ventura, C/C++ - Ed. Petrini