

## 2 Operatori matematici e costrutto `if`

Questa dispensa propone esercizi sulla scrittura di algoritmi, in linguaggio C, utili alla comprensione delle operazioni tra numeri e del costrutto condizionale `if`. Si introducono anche le due funzioni principali della libreria `stdio.h`: `scanf` e `printf`.



Figura 2.1: Un consiglio da seguire.

Nel caso in cui ci fossero delle domande riguardanti il funzionamento dei costrutti di base si fa riferimento alla Figura 2.1. Nello specifico, si può consultare come riferimento:

- Il manuale del corso: “Informatica: arte e mestiere”, D. Mandrioli, S. Ceri, L. Sbattella, P. Cremonesi, G. Cugola, McGraw-Hill Education;
- Un manuale di C on-line: <http://www.cplusplus.com/>.

Si assume una conoscenza di base sul linguaggio C, tale da permettere al lettore di comprendere il significato del seguente frammento di codice.

```
#include <stdio.h>

void main() {

    /* Corpo del programma */

    getchar();
```

```
}
```

Ai fini del corso, è ininfluenza la scelta di dichiarare il `main` come

```
int main(int argc, char *argv[]) {  
  
    /* Corpo del programma */  
  
    getchar();  
    return 0;  
}
```

oppure come

```
void main() {  
  
    /* Corpo del programma */  
  
    getchar();  
}
```

Tuttavia, si richiede che il lettore comprenda la differenza tra le due alternative. La riga di codice `Corpo del programma` non viene considerata dal compilatore, in quanto circondata dai caratteri `/*` e `*/`, che indicano l'apertura e la chiusura di un commento, rispettivamente. Nel caso in cui si voglia commentare da un punto fino alla fine della riga, può essere anche usata l'espressione `//`. L'istruzione `getchar()` non fa parte della soluzione. Si tratta di un'istruzione bloccante per mettere l'elaboratore in attesa di un carattere da tastiera. Senza questa istruzione, o istruzioni equivalenti (e.g., la `system("PAUSE")`), l'esecuzione del programma termina immediatamente senza permettere all'utente di visualizzare l'output a video.

## 2.1 Operazioni matematiche

Si assume che il lettore sia familiare con i tipi di dato numerici previsti dal C (e.g., `int`, `float`) e con i rispettivi di caratteri che di specifica del formato (e.g., `"%d"`, `"%f"`). Inoltre, le operazioni matematiche essenziali necessarie alla comprensione degli esercizi proposti in questa sezione sono:

```
#include <stdio.h>

void main() {
    printf("Addizione: 1+2 = %d\n", 1+2);

    printf("Moltiplicazione: 1*2 = %d\n", 1*2);

    printf("Sottrazione: 1-2 = %d\n", 1-2);

    printf("Divisione: 8/3 = %d (%f)\n", 8/3, 8.0/3.0);

    printf("Resto della divisione intera: 8 mod 3 = %d", 8 % 3);

    getchar();
}
```

Prima di procedere oltre, il lettore deve aver compreso il significato di questo frammento di codice (e.g., provando a compilarlo e ad eseguirlo).

## 2.2 Costrutto *if* e condizioni

Il costrutto *if* codifica un ramo condizionale. Il linguaggio C segue la seguente sintassi:

```
if (condizione)
    statement;
[else statement;]
```

dove le parentesi quadre indicano che la parte **else** *statement*; è opzionale. Come per tutti gli altri costrutti in C, se uno *statement* è una sola istruzione terminata da punto e virgola, non serve altro. Se invece uno *statement* è composto da più istruzioni terminate da punto e virgola, sarà necessario racchiuderlo tra parentesi graffe, ossia:

```
if (condizione) {
    istruzione1;
    istruzione2;
    ...
}
```

La *condizione* è un'espressione booleana, ovvero un'istruzione che, quando valutata, risulta sempre in un valore pari a zero (0, falso) o uno (1, vero). Per comporre espressioni booleane complesse si utilizzano i seguenti operatori:

**Operatori relazionali** valutano relazioni binarie tra i due operandi:

- < minore di
- <= minore di o uguale uguale a
- > maggiore di
- >= maggiore di o uguale a
- == uguale a
- != non uguale a (diverso)

**Operatori booleani** valutano condizioni di verità tra i due operandi

- && AND (congiunzione logica)
- || OR (disgiunzione logica)

**Attenzione:** si osservi che in C, l'operazione di assegnamento  $a = 3$  è diversa dall'operazione di confronto  $a == 3$ . La prima è sempre valutata come vera (1, uno), mentre

la seconda, ovviamente, dipende dal valore memorizzato in *a*. Perciò:

```
#include <stdio.h>

void main() {
    int a;

    scanf("%d", &a); //leggi(a)

    if (a == 4) //confronto
        printf("La variabile 'a' contiene il valore 4\n");
    else
        printf("La variabile 'a' NON contiene il valore 4\n");

    if (a = 4) //assegnamento
        printf("Questo ramo viene sempre eseguito.\n");
    else
        printf("Questo ramo NON viene mai eseguito.\n");

    getchar();
}
```

## 2.2.1 Esercizi

### Esercizio 2.1

Scrivere un programma che esegua la differenza di due numeri interi inseriti da tastiera.

### Esercizio 2.2

Scrivere un programma che riceve in ingresso un prezzo (numero razionale) ed uno sconto (intero tra 0 e 100) da applicare, e restituisce il prezzo scontato e il risparmio ottenuto.

### Esercizio 2.3

Scrivere un programma che prende in ingresso un tempo espresso in ore, minuti e secondi e ne restituisce l'equivalente in secondi.

### Esercizio 2.4

Scrivere un programma che prende in ingresso un tempo espresso in secondi e ne restituisce l'equivalente nel formato ore, minuti, secondi.

### Esercizio 2.5

Scrivere un programma che calcoli la distanza tra due punti,  $a$  e  $b$ , interi sulla retta  $y = 0$ .



### Esercizio 2.6

Scrivere un programma che calcoli la distanza tra due punti,  $a$  e  $b$ , interi su un retta. Potete utilizzare la funzione `abs()` della libreria `math.h`, che calcola il valore assoluto di un numero intero.

```
printf("abs(1-2) = %d", abs(1-2));  
//output: abs(1-2) = 1
```

### Esercizio 2.7

Scrivere un programma che legga da input un numero intero e stampi su output:

- la stringa `basso` se il numero è compreso tra 0 e 3;
- la stringa `Medio` se il numero è compreso tra 4 e 8;
- la stringa `ALTO!` se il numero è compreso tra 9 e 10;
- la stringa `Numero non valido` altrimenti.

### Esercizio 2.8

1. Scrivere un programma che dati tre interi positivi valuti se essi possono essere i lati di un triangolo
2. Nel caso di risposta positiva al punto precedente si comunichi anche il tipo di triangolo (scaleno, isoscele, equilatero, rettangolo)

### Esercizio 2.9

Scrivere un programma che legga da tastiera un numero intero che rappresenta un anno (e.g., 2012) e che determini poi se tale anno è bisestile o meno. Si può assumere che il numero intero letto da tastiera sia sempre valido (e.g., di 4 cifre, positivo).

Un anno è bisestile se è multiplo di 4 ma non di 100, oppure se è multiplo di 400.

## Soluzioni

### Soluzione dell'esercizio 2.1

Risoluzione del problema: dovremo innanzitutto leggere e memorizzare i due numeri. Una volta eseguita l'operazione di differenza dovremo stampare a video il risultato dell'operazione.

```
#include <stdio.h>

// inizio della procedura principale, detta "main"
void main() {
    /* Dichiarazione delle variabili, equivalenti ai "foglietti". */
    int a;
    int b;
    int differenza;

    /* Stampa a video della stringa formattata. Il carattere speciale
       "\n"
       * manda a capo. */

    printf("Inserisci il primo numero \n"); // stampa("...")

    /* Legge un valore di tipo intero (i.e., "%d") e lo scrive nella
       cella di memoria della variabile "a". L'indirizzo di memoria
       di tale cella specificato con l'operatore "&", ovvero "
       indirizzo di" */

    scanf("%d", &a); //leggi(a)

    // Come sopra
    printf("Inserisci il secondo numero \n"); //stampa("...")
    scanf("%d", &b); //leggi(b)

    /* Calcola la differenza tra il valore memorizzato nella
       variabile "a" ed
       * il valore memorizzato nella variabile "b". Il risultato dell'
       operazione
       * scritto nella variabile "differenza". */
    differenza = a - b;

    /* Stampa a video la stringa formattata. Al posto dello
       specificatore di
       * formato "%d" verr stampato il contenuto della variabile "
       differenza",
       * formattato come intero. */
    printf("La differenza %d\n", differenza);
}
```

### Soluzione dell'esercizio 2.2

Risoluzione del problema: leggiamo e memorizziamo la cifra e lo sconto. Dopodichè calcoliamo il prezzo scontato e, facendone la differenza con il prezzo iniziale, calcoliamo l'ammontare dello sconto. Infine stampiamo a schermo i due valori ottenuti.

```
#include <stdio.h>

void main()
{
    float prezzo;
    float sconto;
    float finale;
    float risparmio;

    printf("Inserisci il prezzo originale:\n");
    scanf("%f", &prezzo);

    printf("Inserisci lo sconto da applicare:\n");
    scanf("%f", &sconto);

    risparmio = prezzo * (sconto / 100);
    finale = prezzo - risparmio;

    printf("Il prezzo scontato : %f\n", finale);
    printf("Il risparmio : %f\n", risparmio);
}
```

### Soluzione dell'esercizio 2.3

Risoluzione del problema: leggiamo i tre dati in ingresso e li memorizziamo. Per calcolare i secondi innanzitutto trasformiamo le ore in minuti (moltiplicando per 60) e successivamente i minuti in ore.

```
#import <stdio.h>

void main() {
    int secondi;
    int sec;
    int min;
    int ore;

    printf("Inserisci il numero di ore:\n");
    scanf("%d", &ore);
    printf("Inserisci il numero di minuti:\n");
```



```

scanf("%d", &min);
printf("Inserisci il numero di secondi:\n");
scanf("%d", &sec);

secondi = (ore * 60 + min) * 60 + sec;

printf("%d ore, %d minuti e %d secondi equivalgono a %d secondi\n",
      ore, min, sec, secondi);
}

```

### Soluzione dell'esercizio 2.4

Risoluzione del problema: leggiamo e memorizziamo i secondi da convertire. Dividendo (per intero) per 60 calcoliamo il numero di minuti e, calcolando il resto, otteniamo i secondi nel secondo formato. Ripetendo l'operazione con i minuti riusciamo a trovare il numero di ore e minuti che ci occorrono nel secondo formato. Stampiamo a schermo ore, minuti e secondi.

```

#import <stdio.h>

void main() {
    int secondi;
    int sec;
    int min;
    int ore;

    printf("Inserisci il numero di secondi:\n");
    scanf("%d", &secondi);

    min = secondi / 60;
    sec = secondi - min * 60;

    ore = min / 60;
    min = min - ore * 60;

    printf("%d secondi equivalgono a (h:m:s) %d:%d:%d",
          secondi, ore, min, sec);
}

```

Oppure in maniera più compatta:

```

#import <stdio.h>

void main() {
    int secondi;
    int sec;
    int min;

```

```
int ore;

printf("Inserisci il numero di secondi:\n");
scanf("%d", &secondi);

ore = secondi / 3600;
min = (secondi - ore * 3600) / 60;
sec = secondi - ore * 3600 - min * 60;

printf("%d secondi equivalgono a (h:m:s) %d:%d:%d",
       secondi, ore, min, sec);
}
```

### Soluzione dell'esercizio 2.5

```
#include <stdio.h>

void main() {
    int a, b, lunghezza;

    printf("Inserisci il primo punto: ");
    scanf("%d", &a);

    printf("Inserisci il secondo punto: ");
    scanf("%d", &b);

    lunghezza = a - b;

    if (lunghezza == 0) {
        printf("I due punti coincidono.\n");
    } else {
        if (lunghezza < 0) {
            lunghezza = -lunghezza;
        }

        printf("La lunghezza del segmento : %d\n", lunghezza);
    }

    getchar();
}
```

### Soluzione dell'esercizio 2.6

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

void main() {
```

```
int a;
int b;
int lunghezza;

printf("Inserisci il primo punto: ");
scanf("%d", &a);

printf("Inserisci il secondo punto: ");
scanf("%d", &b);

lunghezza = abs(a - b);

printf("La lunghezza del segmento : %d\n", lunghezza);

getchar();
}
```

### Soluzione dell'esercizio 2.7

```
#include <stdio.h>

void main(){

int numero;

printf("Inserire un numero da 0 a 10: ");
scanf("%d",&numero);

if (numero >= 0 && numero <= 3)
    printf("basso\n");
if (numero >= 4 && numero <= 8)
    printf("Medio\n");
if (numero >= 9 && numero <= 10)
    printf("Alto!\n");
if (numero < 0 || numero > 10)
    printf("Numero non valido\n");

}
```

Oppure nella versione più ottimizzata:

```
#include <stdio.h>

void main(){

int numero;

printf("Inserire un numero da 0 a 10: ");
```

```
scanf("%d",&numero);

if (numero >= 0 && numero <= 3)
    printf("basso\n");
else {
    if (numero >= 4 && numero <= 8)
        printf("Medio\n");
    else {
        if (numero >= 9 && numero <= 10)
            printf("Alto!\n");
        else
            printf("Numero non valido\n");
    }
}
}
```

### Soluzione dell'esercizio 2.8

```
#include <stdio.h>

void main(){

int a,b,c;
int somma_a, somma_b, somma_c;
int diff_a, diff_b, diff_c;

scanf("%d",&a);
scanf("%d",&b);
scanf("%d",&c);

/* somma minore del terzo*/
somma_a = b+c > a;
somma_b = a+c > b;
somma_c = a+b > c;

/* differenza maggiore del terzo*/
diff_a = b+c > a;
diff_b = a+c > b;
diff_c = a+b > c;

printf("I tre numeri");
if (!(somma_a && somma_b && somma_c && diff_a && diff_b && diff_c))
    printf(" non");
printf(" sono i lati di un triangolo\n");

}
```

```
#include <stdio.h>

void main(){

int a,b,c;
int somma_a, somma_b, somma_c;
int diff_a, diff_b, diff_c;
int is_triangolo;
int ipotenusa, somma;

scanf("%d",&a);
scanf("%d",&b);
scanf("%d",&c);

/* somma minore del terzo*/
somma_a = b+c > a;
somma_b = a+c > b;
somma_c = a+b > c;

/* differenza maggiore del terzo*/
diff_a = b+c > a;
diff_b = a+c > b;
diff_c = a+b > c;

is_triangolo = somma_a && somma_b && somma_c && diff_a && diff_b &&
diff_c;
printf("I tre numeri");
if (!is_triangolo)
printf(" non");
printf(" sono i lati di un triangolo");

if (is_triangolo) {
if (a == b && b == c)
printf(" equilatero");
else {
if (a != b && a!= c && c != b)
printf(" scaleno");
else
printf(" isoscele");
// controllo se rettangolo
if (a >= b && a >= c) {
ipotenusa = a * a;
somma = b*b + c*c;
}
if (b > a && b >= c) {
ipotenusa = b * b;
somma = a*a + c*c;
}
if (c >= a && c >= b) {
```

```
        ipotenusa = c * c;
        somma = a*a + b*b;
    }
    if (ipotenusa == somma)
        printf(" (rettangolo)");
}
}
printf("\n");
}
```

### Soluzione dell'esercizio 2.9

```
#include <stdio.h>

void main() {
    int anno;
    int bisestile;

    printf("\nInserisci il numero dell'anno: ");
    scanf("%d", &anno);

        if (anno % 4 == 0) {
            bisestile = 1;
            if (anno % 100 == 0 && anno % 400 !=
                0)
                bisestile = 0;

            else
                bisestile = 0;
        }

    printf("L'anno %d ", anno);
    if (!bisestile)
        printf("non ");
    printf("e' bisestile\n");
}
}
```

oppure

```
#include <stdio.h>

void main() {
    int anno;
    int d4,d100,d400;

    scanf("%d",&anno);
```

```
d4 = anno % 4 == 0;
d100 = anno % 100 == 0;
d400 = anno % 400 == 0;

if ((d4 && !d100) || d400)
    printf("E' bisestile\n");
else
    printf("Non e' bisestile\n");
}
```