1 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 1.1: Un consiglio da seguire.

Nel caso in cui ci fossero delle domande riguardanti il funzionamento dei costrutti di base si fa riferimento alla Figura 1.1. 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();
```

```
}
```

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. Nel caso utilizziate Code-Blocks l'istruzione per bloccare il programma alla fine dell'esecuzione non è necessaria in quanto tale programma attende che l'utente prema un tasto prima di terminare l'esecuzione del codice.

1.1 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).

1.2 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();
}
```

1.2.1 Esercizi

Esercizio 1.1

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

Esercizio 1.2

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 1.3

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

Esercizio 1.4

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

Esercizio 1.5

Scrivere un programma che prende in ingresso un prezzo in euro e restituisce il numero minimo di banconote utilizzando solo pezzi da 50, 20 e 5 euro. Indicare anche la moneta rimanente.

Esercizio 1.6

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

Esercizio 1.7

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 1.8

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 1.9

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 1.10

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.

Esercizio 1.11

1. Scrivere la tabella di veritá della proposizione $A \Rightarrow B$ (implicazione logica), sapendo che essa è equivalente alla proposizione $!A \,||\, B$ (due proposizioni sono equivalenti se la loro tabella di verità è identica).

2. Scrivere la tabella di verità della proposizione $A \Leftrightarrow B$ (co-implicazione logica), sapendo che essa è equivalente alla proposizione $A \Rightarrow B \;\&\&\; B \Rightarrow A$.

3. Mostrare che la proposizione $A \Rightarrow B$ non è equivalente alla proposizione $A \,||\, !B$.

Esercizio 1.12

(TdE Gennaio 2019) Non vado in vacanza se almeno due dei tre seguenti predicati non sono soddisfatti: (A) ho un gruzzoletto da parte, (B) ho almeno due settimane in cui posso non studiare, (C) ho finito gli esami a luglio.

1. Scrivere la tabella di verità per il predicato "Non vado in vacanza" attraverso i predicati semplici A, B e C.

2. La traduzione di tale condizione con $((!A \,||\, !B) \;\&\&\; (!A \,||\, !C) \;\&\&\; (!B \,||\, !C))$ è corretta?

Esercizio 1.13

Assumendo che `c1` e `c2` siano due variabili di tipo char, che memorizzano rispettivamente i valori `'e'` ed `'m'` indicare, per ognuna delle espressioni logiche:

1. se l'espressione è vera o falsa (per i valori delle variabili sopra indicati);

2. se è sempre vera per qualsiasi valore che le due variabili possono assumere;

3. se è sempre falsa per qualsiasi valore che le due variabili possono assumere.

Si fornisca una giustificazione per ogni risposta inserita nella tabella (risposte senza giustificazione potranno essere considerate nulle).

1. `((c1 != 'e') && (c2 == 'm')) || ((c1 != 'h') && (c2 == 'm'))`

2. `(c1 < 'g') || !((c1 <= 'g') && (c1 != 'g'))`
3. `(c1 <= 'm') || ((c2 > 'm') || !(c2 > c1))`

Esercizio 1.14

(TdE Novembre 2014) Si consideri la seguente espressione booleana:

$$((!A \,||\, B) \,\&\&\, (A \,\&\&!B)) \,||\, C$$

e se ne compili la corrispondente tabella di verità (ossia una tabella in cui 0 rappresenta il valore logico FALSO, 1 il valore VERO).

Si consideri ora la condizione, scritta in linguaggio C, in cui x e y siano due variabili intere:

$$((!(x < 0)||(y > 0)) \,\&\&\, ((x < 0) \,\&\&\, !(y > 0)))) \,||\, (x > 3)$$

ottenuta dall'espressione booleana sopra indicata sostituendo la variabile A con $x < 0$, la variabile B con $y > 0$, la variabile C con $x > 3$ Si risponda alle seguenti domande giustificando la risposta:

1. L'espressione è vera o falsa quando $x = 5$ e $y = 7$?
2. Se $x < 0$, per quali valori di y l'espressione è vera?

Esercizio 1.15

(TdE Luglio 2018) Si considerino le seguenti frasi, esprimibili in logica proposizionale tramite tre predicati (A, B, C), che esprimono due condizioni per cui si può entrare al cinema gratis:

1. Sei il millesimo cliente E sei il cliente più paffutello della sala, OPPURE non hai mai visto il film in programma.
2. Hai già visto il film in programma E almeno una delle seguenti condizioni è vera:
   - non sei il millesimo cliente;
   - c'è qualche cliente in sala paffutello almeno quanto te.

Si definiscano i tre predicati logici A, B, C contenuti nelle due precedenti frasi, si traducano quindi le due frasi in espressioni booleane, si dica se le proposizioni espresse nelle due frasi sono equivalenti, giustificando la risposta.