2 Costrutti `while`, `for` e `switch`

Questa dispensa propone esercizi sulla scrittura di algoritmi, in linguaggio C, utili alla comprensione dei costrutti `while`, `for` e `switch`.

I costrutti per costruire cicli in C sono il `while`, la variante `do...while` e il `for`.

```
inizializzazione; //opzionale

while (condizione) {
  corpo;

  incremento; //opzionale
}
```

La `condizione` è valutata prima di ogni iterazione, inclusa la prima; quindi il corpo del ciclo (e quindi anche l'eventuale istruzione di `incremento`) potrebbero non eseguire mai, nel caso in cui la condizione sia falsa dall'inizio.

```
inizializzazione; //opzionale

do {
  corpo;

  incremento; //opzionale
} while (condizione);
```

Invece, nella variante `do...while`, il corpo e l'eventuale istruzione di incremento sono eseguiti almeno una volta prima di valutare la condizione.

Il `for` è equivalente, ma ha una sintassi più compatta:

```
for (inizializzazione; condizione; incremento) {
  corpo;
}
```

Si noti che le parentesi sono necessarie solo nel caso di corpo con più di un'istruzione. Inoltre, l'espressione di inizializzazione e di incremento sono opzionali. Di fatto, un ciclo `while` può essere scritto in modo equivalente con un costrutto `for` nel seguente modo:

```
inizializzazione;

for (; condizione; ) {
  corpo;

  incremento;
}
```

Il costrutto del C che permette di scegliere tra alternative multiple (più di 2) è lo `switch`.

```
inizializzazione;

switch (variabile) {
  case valore1:
      istruzione1;
      break;
  case valore2:
      istruzione2;
      break;
  default:
      istruzione3;
      break;
}
```

Il `break` tra un case e l'altro permette che venga eseguito solo il codice relativo al case in cui si è capitati. Il `break` finale non è necessario ma consigliato (nel caso in cui si dovesse immettere nuovi case).

2.0.1 Esercizi

Esercizio 2.1

Scrivere un programma che dato un numero positivo ne restituisca la radice intera

Esercizio 2.2

1. Scrivere un programma che dato un numero reale positivo l e un intero positivo n restituisca l'area del poligono regolare con n lati di lunghezza l. Si implementi una soluzione per che gestisca i casi $n \in \{1, \ldots, 6\}$ e che sia espandibile agevolmente. Suggerimento: l'area del pentagono regolare è:

$$A = l^2 \frac{5}{2} \sqrt{\frac{\sqrt{5}}{10} + \frac{1}{4}} \tag{2.1}$$

2. Supporre ora che l'utente possa inserire valori negativi per l e n. Si ripeta l'acquisizione dei dati finché l'utente non inserisce dei valori accettabili per i due parametri.

Esercizio 2.3

Scrivere un programma che richiede all'utente un intero positivo e ne stampa a schermo tutti i divisori.

Esercizio 2.4

Scrivere un programma che richiede all'utente un intero positivo e determina se è primo o meno. Il programma deve continuare a chiedere il numero fino a che l'utente non ne inserisce uno positivo.

Esercizio 2.5

Scrivere un programma che richiede all'utente un intero positivo N e stampa a schermo i primi N numeri primi. Ad esempio: con $N = 7$ a schermo avremo 2 3 5 7 11 13 17.

Esercizio 2.6

Scrivere un programma che richiede all'utente un indovinare un numero casuale (generato con la funzione `rand`) tra 1 e 10. Il gioco si svolge nel seguente modo: se l'utente ha indovinato allora viene stampato a schermo il successo, altrimenti il programma dovrà guidare l'utente dandogli come suggerimento se il numero segreto è più grande o più piccolo di quello inserito.

Esercizio 2.7

Dati due numeri m e n questi sono numeri amicali se la somma dei divisori di m è uguale a n, e viceversa (per esempio 220 e 284, 1184 e 1210, 2620 e 2924, 5020 e 5564, 6232 e 6368, 17296 e 18416). Ad esempio si ha:

$$220 \rightarrow 110 + 55 + 44 + 22 + 20 + 11 + 10 + 5 + 4 + 2 + 1 = 284$$
$$284 \rightarrow 142 + 71 + 4 + 2 + 1 = 220$$

Scrivere un programma che ricevuto in ingresso due numeri interi restituisce 1 se i numeri sono amicali, 0 altrimenti.

Esercizio 2.8

Scrivere un programma che dato un numero reale x e un intero positivo n calcoli lo sviluppo di McLaurin del seno in x all'ordine n. Si ricorda che lo sviluppo di Taylor del seno è dato dalla sommatoria:

$$\sin x = \sum_{i=0}^{n} \frac{(-1)^i}{(2i+1)!} x^{2i+1}$$