ESERCIZI DI PROGRAMMAZIONE

Prof. Marco Camurri

Capitolo 1 - Strutture di controllo

Sequenza e selezione

0. Hello World!

Scrivi un programma che visualizza prima la frase "Hello World!", e poi le parole "Hello" e "World" su due diverse righe.

---

1. Area del cerchio

Scrivi un programma che calcola l'area del cerchio in base al valore del raggio.

---

2. Valore assoluto

Scrivi un programma che richiede all'utente un numero reale e ne calcola il valore assoluto. Ricorda che se un numero è maggiore o uguale a zero, il suo suo valore assoluto è uguale al numero stesso, mentre se il numero è negativo il suo valore assoluto si ottiene cambiando il segno del numero.

---

3. Maggiorenni e minorenni

Scrivi un programma che stabilisce se una persona è maggiorenne o minorenne in base all'età.

---

4. Angoli

Scrivi un programma che richiede all'utente la misura, in gradi, dell'angolo di un triangolo e comunica se l'angolo è retto, acuto o ottuso. La misura inserita dall'utente deve essere un numero compreso tra 0 e 180. In caso contrario terminare il programma con il messaggio "valore non valido".

---

5. Classificazione triangoli

Scrivi un programma che richiede all'utente le misure dei tre lati di un triangolo, e comunica se il triangolo è equilatero, isoscele o scaleno.

---

6. Ordinamento di tre valori

Scrivi un programma che richiede all'utente tre numeri reali e li visualizza in ordine crescente. Esempio:

```
Inserisci tre numeri: 8 4 10
I tre numeri in ordine sono: 4 8 10
```

Suggerimento: dati tre valori, esistono sei possibili modi in cui questi possono essere ordinati: abc, acb, bac, bca, cab, cba.

---

7. Equazioni di secondo grado

Scrivi un programma che risolve equazioni di secondo grado a coefficienti reali, $ax^2 +bx +c = 0$. Il programma deve segnalare opportunamente il caso di equazione impossibile. Si ricorda che le due soluzioni sono date dalla formula: $(-b \pm \sqrt{b^2 - 4ac})/(2a)$. L'equazione è impossibile se la quantità $b^2 - 4ac$, detta delta, è negativa. Ricorda che per il calcolo della radice quadrata occorre usare la funzione sqrt() della libreria cmath (C++) o math.h (C).

---

8. Formula di Erone

Dato un triangolo qualsiasi i cui lati misurano a, b e c, la formula di Erone afferma che l'area del triangolo è data da $\sqrt{p(p-a)(p-b)(p-c)}$, dove p è il semiperimetro, ovvero la metà del perimetro del triangolo. Scrivi un programma che consenta di calcolare l'area di un triangolo date le misure dei lati. Prima di calcolare l'area, il programma deve verificare che le misure inserite dall'utente siano tutte positive, e che i tre lati formino effettivamente un triangolo (in un triangolo ogni lato deve essere minore della somma degli altri due). In caso contrario, il programma deve richiedere nuovamente l'inserimento dei valori di input. Ricorda che per il calcolo della radice quadrata occorre usare la funzione sqrt() della libreria cmath (C++) o math.h (C).

Cicli while e for

9. N volte CIAO

Scrivi un programma che richiede all'utente un numero intero positivo N, e visualizza N volte la stringa "Ciao!" sul video, dopodiché visualizza una sola volta la stringa "Arrivederci". Esempio:

```
Quante volte vuoi ripetere la parola? 3
Ciao!
Ciao!
Ciao!
Arrivederci.
```

---

10. Pari e dispari

Scrivi un programma che mostra i primi N numeri naturali, con N fornito dall'uten. Esempio:

```
Quanti numeri vuoi mostrare? 5
1 dispari
2 pari
3 dispari
4 pari
5 dispari
FINE.
```

---

11. Elenco dei divisori di un numero

Scrivi un programma che mostra tutti i divisori di un numero naturale N fornito dall'utente. Esempio:

```
Inserisci un numero intero: 10
I divisori di 10 sono: 1, 2, 5, 10.
```

---

12. Conto alla rovescia

Scrivi un programma che visualizza una sequenza decrescente di numeri interi, a partire dal numero fornito dall'utente e fino al valore 0. Al termine della sequenza il programma deve mostrare il messaggio FINE! . Ad esempio, se l'utente fornisce in input il valore 5, il programma stampa:

```
5, 4, 3, 2, 1, 0, FINE!
```

---

13. Tabelline

Scrivi un programma che mostra i primi K multipli di un numero interno N. I valori di N e K sono forniti dall'utente. Ad esempio, per N=3 e K=5 il programma deve mostrare il seguente output:

```
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
```

---

14. Somma dei primi N numeri naturali

Scrivi un programma che richiede all'utente un numero intero positivo N, calcola la somma dei numeri naturali compresi tra 1 ed N. Ad esempio, se l'input è N=5, l'output è 1+2+3+4+5=15.

---

15. Fattoriale

Dato un numero naturale N, si definisce "fattoriale di N" il prodotto di tutti i numeri naturali compresi tra 1 ed N stesso. Ad esempio, il fattoriale di 5 è 120, poiché 120=1*2*3*4*5. Scrivi un programma che calcola il fattoriale del numero fornito dall'utente.

---

16. Quadrati dei numeri dispari

Scrivi un programma visualizza i quadrati dei primi N numeri dispari, e ne calcola la somma. Il valore di N è fornito dall'utente.

---

17. Somma dei numeri pari

Scrivi un programma che calcola la somma di tutti i numeri pari compresi tra A e B, estremi inclusi. A e B sono due numeri interi positivi forniti dall'utente, con A < B. Esempio: INPUT: A=2, B=11; OUTPUT: 30.

---

18. Numeri armonici

Scrivi un programma che richiede all'utente un numero intero positivo N e calcola la somma dei reciproci dei primi N numeri naturali, ovvero: 1 + 1/2 + 1/3 + ... + 1/N. In matematica, il numero razionale ottenuto in tal modo è detto n-esimo numero armonico. Ad esempio, il decimo numero armonico è: 1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + 1/7 + 1/8 + 1/9 + 1/10 , circa pari a 2.93.

---

19. Serie armonica alternata

Scrivi un programma che richiede all'utente un numero intero positivo N e calcola la quantità: 1 – 1/2 + 1/3 – 1/4 ... +/- 1/N. Curiosità: al crescere di N, il numero ottenuto si avvicina sempre più al logaritmo naturale di 2.

---

20. Numeri perfetti

Un numero naturale si dice perfetto se è uguale alla somma dei suoi *divisori propri* (per divisore proprio intendiamo un divisore di N diverso da N stesso). Ad esempio, il numero 28 ha come divisori propri i numeri 1, 2, 4, 7 e 14, ed è un numero perfetto poiché 1 + 2 + 4 + 7 + 14 = 28. Scrivi un programma che richiede all'utente un numero e stabilisce se questo è perfetto oppure no. Curiosità: i numeri perfetti sono piuttosto rari. I primi numeri perfetti sono 6, 28, 496... qual'è il prossimo? Come si potrebbe modificare il programma in modo da mostrare tutti i numeri perfetti compresi tra 1 e 1.000.000.000 ?

---

21. Numeri primi

Un numero naturale si dice primo se è divisibile solo per 1 e per sè stesso. Scrivi un programma che stabilisce se un numero fornito dall'utente è primo oppure no. Cerca di ottimizzare il programma in modo che venga effettuato il minor numero di operazioni.

---

22. Fizz Buzz

In alcuni paesi, per insegnare ai bambini i multipli di tre e di cinque, si utlizza un gioco di gruppo chiamato "Fizz Buzz". I bambini si siedono in cerchio e devono contare ad alta voce, a turno, partendo da 1, sostituendo ogni multiplo di tre con la parola "Fizz", ogni multplo di 5 con la parola "Buzz" e ogni multiplo sia di tre che di cinque con la parola "Fizz Buzz". Ad esempio, se si decide di contare fino a 15, il primo bambino dice "1", il secondo bambino dice "2", il terzo bambino dice "fizz", il quarto dice "4" e così via. Scrivi un programma che mostra la sequenza corretta fino al numero N fornito dall'utente. Ad esempio, se N=16 il programma mostra:

```
1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14,
Fizz Buzz, 16,
```

---

23. Potenze

Scrivi un programma che calcola la potenza a^b, dove a e b sono due interi positivi forniti dall'utente. Esempio: a = 2, b = 3 --> output = 8.

---

24. Media aritmetica

Scrivi un programma che richiede all'utente N numeri reali, e mostra la media aritmetica dei numeri inseriti. Il valore di N è deciso preventivamente dall'utente. Importante: per raccogliere i valori inseriti dall'utente, usa una variabile adatta a contenere numeri con la virgola. Esempio di esecuzione del programma:

```
Quanti numeri vuoi inserire? 3
Dammi un numero(1): 5
Dammi un numero(2): 6.5
Dammi un numero(3): 6
La media e' 5.83333
```

Ciclo do-while e controllo dell'input

25. Controllo dell'input

Scrivi un programma che richiede all'utente un numero MAGGIORE DI 100 e PARI . Se il numero inserito non e' valido, il programma deve richiederlo nuovamente. Se il numero e' valido, il programma termina. Risolvi l'esercizio usando un ciclo do-while. Esempio di esecuzione:

```
Inserisci un numero pari e maggiore di 100: 20
Hai inserito un numero non valido
Inserisci un numero pari e maggiore di 100: 103
Hai inserito un numero non valido
Inserisci un numero pari e maggiore di 100: 102
Bravo. Fine.
```

---

26. Controllo dell'input (2)

Un distributore di merendine accetta solo monete da 5, 10, 20 e 50 centesimi. Scrivi un programma che richiede il valore di una moneta, e la accetta solo se la moneta ha un valore valido. Risolvi l'esercizio usando un ciclo do-while. Esempio di esecuzione:

```
Inserisci una moneta: 30
Moneta non valida
Inserisci una moneta: 55
Moneta non valida
Inserisci una moneta: 20
MONETA ACCETTATA. Fine.
```

---

27. Conto alla rovescia con ciclo do-while

Riscrivi il programma del conto alla rovescia usando un ciclo do-while al posto del ciclo while.

---

28. Distrubutore di bibite con resto

Un distributore di bibite permette di acquistare le seguenti bevande:

1. bottiglietta d'acqua (costo: 50 cent)
2. bibite in lattina (costo: 80 cent)
3. bibite in bottiglietta (costo: 1 euro = 100 cent)

Il distrubitore accetta tutte le monete (1 cent,5 cent, 10 cent, 20 cent, 50 cent, 1 euro, 2 euro). Il funzionamento del distributore è il seguente: L'utente seleziona il prodotto (1=acqua, 2=lattina, 3=bottoglia) Il distributore richiede monete finch` l'importo inserito non risulta maggiore o uguale all'importo richiesto. Terminato l'inserimento delle monete, il distributore eroga il prodotto e visualizza sul display il resto da ritirare.
Scrivi un programma C++ che simula il funzionamento del distributore. Note: per simulare l'inserimento delle monete da 1 e 2 euro, l'utente del programma inserisce i valori 100 e 200. Anche il resto viene mostrato in centesimi (ad esempio, il resto di 1 euro e 20 cent viene mostrato come 120).

Cicli annidati

29. Ciclo annidato - determinazione output

Determina l'output dei seguenti frammenti di programma che contengono cicli annidati:

```
// A)
int i=0,j,R=3,C=4;
while( i<R ) {
      j=0;
      while ( j<C ) {
              cout << i+j;
              j=j+1;
      }
      cout << "\n";
      i=i+1;
}

// B)
int x=0,y=0;
while( x<10 )
{
      y=0;
      while ( y<10 )
```

```
      {
        if ( x<y )
          cout << "+";
        else
          cout << "-";
        y=y+1;
      }
      cout << "\n";
      x=x+1;
}

// C
int r=1,c=1;
while( r<=6 ) {
      c=1;
      while ( c<=r ) {
        cout << r;
        c=c+1;
      }
      cout << "\n";
      r=r+1;
}
```