Laboratorio di Fondamenti di Informatica

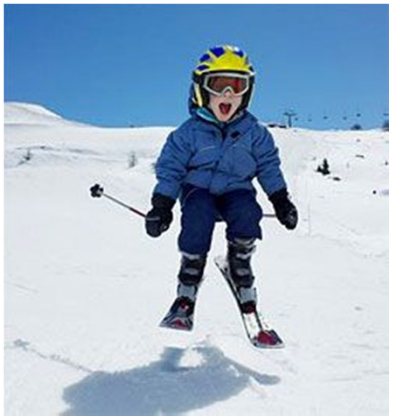
Esercizi di "familiarizzazione"

Lab01_2015-10-15

Anno accademico 2015/2016

Da sapere, prima di lanciarsi…

Proveremo a classificare gli esercizi in base alla loro "difficoltà rispetto al momento in cui sono assegnati":



1. **Esercizio** che non pone alcun problema, e richiede concetti già "consolidati" nel corso. Risolverlo "a colpo sicuro" dovrebbe indicare che si è "al passo" col programma.

2. **Esercizio** che richiede più attenzione, più tempo, e magari qualche concetto appena introdotto.

3. **Esercizio** che richiede confidenza nelle proprie capacità e dimestichezza con tutti gli strumenti. E, forse, a un po' di inventiva, coraggio, creatività.

N.B.: il tentativo di classificazione fa riferimento al momento di assegnazione e alle conoscenze di chi parte da zero: gli esercizi neri di oggi saranno rossi domani e azzurri dopodomani!

1. 356,25 (circa)

Scrivere un programma che, dato un anno inserito dall'utente come numero intero (ad esempio 2014), dica se è bisestile o meno. Esempio:

```
Quale anno ti interessa? 1777
Anno NON bisestile
```

Si ricorda che, a partire dal 1582, un anno è considerato bisestile se è multiplo di 4 (e.g., il 2012). Se però è un multiplo di 100 (e.g., il 1900), allora non è bisestile. È tuttavia bisestile se è anche multiplo di 400 (e.g., il 2000).

Più sinteticamente, gli anni successivi alla bolla papale "Inter gravissimas" sono bisestili se divisibili per 4, con l'eccezione di quelli secolari non divisibili per 400.

Suggerimenti:

- Usare **scanf()** per leggere il valore da analizzare
- Utilizzare l'operatore **%** per calcolare il resto della divisione intera
- Si possono usare più istruzioni condizionali (**if**…) annidate, oppure si possono combinare opportunamente le condizioni con AND e OR

2. Sei Perotto!

2. Scrivere un programma per il "ripasso delle tabelline".
Il programma riceve dall'utente (su stdin) un numero n compreso tra 2 e 12, e stampa la tabellina di n nel modo seguente. Ad esempio, per n = 4

```
Tabellina da ripassare:  4
        4 x 1 = 4
        4 x 2 = 8
        …
        4 x 10 = 40
```

Varianti e aggiunte

Si ricevano due interi n e k, e si stampi la «matrice» di tutte le tabelline da 1 a n, limitandosi al k-esimo valore. Ad esempio, per n=5 e k = 9:

```
     1   2   3   4   5   6   7   8   9
  ---------------------------------------
2 :  2   4   6   8  10  12  14  16  18
3 :  3   6   9  12  15  18  21  24  27
4 :  4   8  12  16  20  24  28  32  36
5 :  5  10  15  20  25  30  35  40  45
```

3. Divisori & 4. Numeri primi

3. Scrivere un programma che dato un numero n intero positivo ne stampa a video tutti i divisori maggiori di 1 e minori di n. Esempo:

```
Inserisci un numero:  20
I divisori sono:  2  4  5  10
```

4. Scrivere un programma che dato un numero intero positivo controlla se il numero è primo o no. Esempio:

```
Inserisci un numero:  101
Si tratta di un numero primo
```

Per comodità di riferimento e controllo, i numeri primi sono:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347…

3&4. Il programma legge **due** interi positivi, ne stampa tutti i divisori comuni maggiori di 1, e controlla se sono primi fra loro. Esempio:

```
Inserisci due numeri:  18  72
I divisori comuni sono:  2  3  6  9  18
I due numeri NON sono primi fra loro
```

5. Sequenze di numeri

Scrivere un programma che chiede all'utente di inserire una sequenza (di lunghezza arbitraria) di numeri interi positivi, terminata da 0, e mentre legge i vari numeri da stdin conta quanti sono i numeri pari e quanti quelli dispari nella sequenza. Alla fine della sequenza, poi, stampa a video il massimo, il minimo, e la media. Esempio:

```
21   3   8   4   5   1   17   0

pari: 2
dispari: 5
minimo = 1
massimo = 21
media = 8.43
```

6. Numeri affiatati (e primi)

Definiamo due numeri ***affiatati*** se sono ***diversi tra loro*** e la loro ***somma*** è pari al ***prodotto delle cifre*** che li compongono.
Ad esempio (14,34) e (63, 81) sono coppie di numeri affiatati:

14 + 34 = 1 x 4 x 3 x 4 = 48
63 + 81 = 6 x 3 x 8 x 1 = 144

6a. Si scriva un programma che elenca tutte le diverse coppie di numeri affiatati composti di due cifre.

6b. Si scriva un programma che elenca tutte le diverse coppie di numeri affiatati composti di due cifre in cui il maggiore dei due numeri è un numero primo (eventualmente, si riusi il codice dell'esercizio **4** !).