

Prima prova parziale di Programmazione I

24 gennaio 2012 (tempo disponibile: 2 ore)

Esercizio 1

(9 punti)

Si scriva un programma `integrale.c` che definisce una funzione

```
double integra(double da, double a, double dx)
```

che restituisce un'approssimazione di

$$\int_{da}^a \sin(x) dx$$

calcolata facendo iterare una variabile `x` dal valore `da` al valore `a` a passi di `dx` e sommando ogni volta il valore del seno a `x` per il valore di `dx`.

Tale programma deve avere anche un `main()` che legge `da`, `a` e `dx` da tastiera (di tipo `double`: si usi il formato `%lf` per `scanf()`) e quindi stampa a video l'approssimazione dell'integrale.

Se tutto è corretto, tale programma si deve comportare ad esempio come segue:

```
$ ./a.out
da: 0
a: 3.1415
dx: 0.2
integrale di sin(x) tra 0.000000 e 3.141500 d0.200000 = 1.997467
```

```
$ ./a.out
da: 3.1415
a: 6.283
dx: 0.001
integrale di sin(x) tra 3.141500 e 6.283000 d0.001000 = -2.000000
```

Esercizio 2

(6 punti)

Si scriva il file `next_prime.c` che definisce la funzione `next_prime()`. Questa funzione deve restituire un diverso numero primo ad ogni chiamata, dal 2 in poi. Solo `next_prime()` deve essere visibile all'esterno, non altre funzioni ausiliare che potreste scrivere. Scrivete poi il file `next_prime.h` che esporta la segnatura della funzione `next_prime()`.

Se tutto è corretto, un'esecuzione del seguente programma:

```
#include <stdio.h>
#include "next_prime.h"

int main(void) {
    int number;
    int c;

    printf("quanti numeri primi vuoi stampare? ");
    scanf("%d", &number);
```

```

    for (c = 0; c < number; c++)
        printf("%d\n", next_prime());

    return 0;
}

```

potrebbe essere:

```

quanti numeri primi vuoi stampare? 7
2
3
5
7
11
13
17

```

Esercizio 3

(16 punti)

Si scriva un programma `numeri_italiani.c` che definisce le funzioni:

- `leggi()`, che legge da tastiera un numero intero non negativo e lo restituisce. Se fosse negativo, deve continuare a chiederlo all'utente;
- `stampa(int numero)`, che stampa le cifre del numero indicato, in italiano. Per esempio, se `numero` è 4301 allora deve stampare `quattro tre zero uno`; se `numero` è 0 allora deve stampare `zero`.

È possibile definire ulteriori funzioni ausiliarie, se servono. La funzione `stampa()` **deve essere ricorsiva o chiamare una vostra funzione ricorsiva**. Si definiscano gli argomenti delle funzioni come `const`, quando possibile.

Infine, tale programma deve avere anche un `main()` che chiama `leggi()` per leggere un numero non negativo e poi chiama `stampa()` per stamparne le cifre in italiano.

Se tutto è corretto, il programma si deve comportare ad esempio come segue:

```

$ ./a.out
inserisci un numero: 10985
uno zero nove otto cinque

$ ./a.out
inserisci un numero positivo: -13
inserisci un numero positivo: 8901
otto nove zero uno

$ ./a.out
inserisci un numero: 300896
tre zero zero otto nove sei

$ ./a.out
inserisci un numero: 0
zero

$ ./a.out
inserisci un numero: 0006
sei

```