

Prima prova parziale di Programmazione I

7 febbraio 2014 (tempo disponibile: 2 ore)

Esercizio 1

(10 punti)

Si scriva una funzione

```
double e(int precision)
```

che restituisce un'approssimazione della costante di Eulero e calcolata come:

$$\sum_{n=0}^{\text{precision}} \frac{1}{n!}$$

Si scriva quindi un `main` che chiede all'utente `precision` e stampa l'approssimazione di e calcolata con la precedente funzione e la precisione inserita, usando 20 cifre decimali. Per esempio, una possibile esecuzione di tale programma potrebbe essere:

```
$ ./a.out
precision: 7
2.7182539682539683668
```

Esercizio 2

(10 punti)

Si scriva una funzione

```
void swap(int array[], int length)
```

che riceve come parametri un array di lunghezza `length`, considera l'array come una sequenza di triplette di interi e, per ogni tripletta, inverte il primo e il terzo elemento della tripletta. Se alla fine rimangono 1 o 2 elementi, non vengono modificati.

Se tutto è corretto, l'esecuzione del seguente `main`:

```
int main(void) {
    int arr[] = { 13, 0, 34, -5, -6, 10, 34, -28, 44, 71, 9 };
    int pos;

    swap(arr, 11);
    for (pos = 0; pos < 11; pos++)
        printf("%i ", arr[pos]);
    printf("\n");
    return 0;
}
```

stamperà

```
34 0 13 10 -6 -5 44 -28 34 71 9
```

Esercizio 3

(12 punti)

Si scriva un programma `primes.c` che definisce la funzione **ricorsiva**

```
int count_primes(int max)
```

che restituisce la quantità di numeri primi tra 0 e `max` inclusi. Si scriva quindi un `main` che legge `max` da tastiera e stampa quanti numeri primi ci sono tra 0 e `max` inclusi. Per esempio, una possibile esecuzione del programma potrebbe essere:

```
$ ./a.out
```

```
max: 13
```

```
Ci sono 6 numeri primi tra 0 e 13
```