

Laboratorio di Programmazione: Linguaggio C

Lezione 12 del 16 gennaio 2014

Damiano Macedonio

Esercizio 1 [Tratto dalla prova parziale del 24 gennaio 2012]

Si scriva un programma C in cui si definisce la funzione `int next_prime(int last)` che restituisce il numero primo immediatamente successivo a `last`.

Si scriva inoltre una funzione `main` che chiede all'utente un valore `p` e chiama la funzione `next_prime` per stampare i primi `p` numeri primi.

Esempio

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

Esercizio 2 [Tratto dalla prova parziale del 4 febbraio 2013]

Si scriva una funzione

```
double pi(int precision)
```

che restituisce un'approssimazione di π calcolata come:

$$\sum_{k=0}^{\text{precision}} \frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right)$$

Si scriva quindi una funzione `main` che chiede all'utente la precisione e stampa l'approssimazione di π calcolata usando la precedente formula con la precisione inserita, usando 30 cifre decimali dopo la virgola.

Esempio

```
$ ./a.out
precision: 6
3.141592653228087783645605668426
```

Esercizio 3 [Tratto dalla prova parziale del 24 gennaio 2012]

Si scriva un programma C in cui si 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 di x moltiplicato per il valore di dx .

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

Per conoscere il seno di x , usare la funzione `double sin(double x)`:

- nel preambolo includere la libreria `math.h` (con `#include <math.h>`)
- per la compilazione usare l'opzione `-lm` (esempio: `gcc integra.c -o integra -lm`)

Esempi

```
$ ./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 4 [Tratto dalla prova parziale del 4 febbraio 2013]

Si scriva un programma C che definisce la funzione

```
int cifra_massima(int num)
```

la quale deve restituire la cifra massima nella rappresentazione decimale di `num`. Tale programma dovrà inoltre definire un `main` che (1) chiede all'utente di inserire un numero non negativo, (2) chiama la funzione `cifra_massima` per calcolarne la cifra massima e (3) stampa tale cifra massima trovata.

Esempi

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

```
Inserisci un numero non negativo: 1232
```

```
La cifra massima di 1232 e' 3
```

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

```
Inserisci un numero non negativo: 0
```

```
La cifra massima di 0 e' 0
```

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

```
Inserisci un numero non negativo: -5
```

```
Inserisci un numero non negativo: 30756
```

```
La cifra massima di 30756 e' 7
```

Esercizio 5 [Tratto dalla prova parziale del 24 gennaio 2012]

Si scriva un programma C in cui sono definite le funzioni:

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

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

Suggerimento: Potete usare l'approccio descritto nell'Esercizio 7 della Lezione 8, oppure utilizzare un array di supporto:

- Per recuperare le cifre che compongono il numero intero utilizzare il modulo della divisione per 10 (`n % 10` restituisce la cifra meno significativa di `n`).
- Memorizzare le cifre in un array di interi di dimensione massima 64.
- Scorrere l'array dall'ultima cifra inserita fino alla cifra in posizione 0 per recuperare le cifre da sinistra a destra.

Esempi

```
$ ./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
```