

INFORMATICA – PROVA SCRITTA, 21 SETTEMBRE 2020

ESERCIZIO 1

Lo *sviluppo di Maclaurin* della funzione $\sin(x)$ è definito come

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots = \sum_{i \geq 0} (-1)^i \frac{x^{2i+1}}{(2i+1)!}. \quad (1)$$

Tale sviluppo può essere utilizzato per calcolare approssimazioni di $\sin(x)$. In particolare, l'*approssimazione di $\sin(x)$ calcolata all' n -esimo termine* si ottiene sommando algebricamente i primi n termini della serie alla destra dell'uguale della formula (1). Ad esempio, l'approssimazione di $\sin(x)$ calcolata al quarto termine si ottiene sommando $x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}$.

Scrivete la funzione `float seno(float x, int n)` che restituisce l'approssimazione di $\sin(x)$ calcolata all' n -esimo termine.

Suggerimento: realizzate a parte una funzione che restituisca il fattoriale di un intero positivo passato per argomento.

ESERCIZIO 2

Sfruttando la funzione `seno` scritta al punto precedente, scrivete un frammento di codice che:

- chieda all'utente un numero reale `x` e un intero `n` e stampi l'approssimazione di $\sin(x)$ calcolata all' n -esimo termine. Tale processo di acquisizione di `x` e `n` e stampa dell'approssimazione di $\sin(x)$ continua finché l'utente inserisce il numero 0 sia per `x` che per `n`;
- all'uscita del ciclo al punto precedente, il programma deve stampare il numero `x` inserito che massimizza il valore assoluto $|x - \sin(x)|$, dove con $\sin(x)$ si intende l'approssimazione di $\sin(x)$ scelta dall'utente all'atto dell'immissione di `x`.

[Altri esercizi alla prossima pagina]

ESERCIZIO 3

Scrivete la funzione

```
void sort_odd( int *A, int d )
```

che ordina in senso crescente gli interi dispari contenuti in **A** e lascia al loro posto gli interi pari di **A**. Potete assumere che l'array passato a `sort_odd` contenga esclusivamente interi positivi. La scrittura di tale funzione non deve prevedere alcun array d'appoggio.

ESERCIZIO 4

Sfruttando la funzione scritta al punto precedente, scrivete un frammento di codice che:

- chieda all'utente 100 interi positivi e li memorizzi in un array **X** nell'ordine in cui sono stati inseriti (in caso di inserimento di un intero non positivo, il programma deve richiedere l'intero all'utente),
- stampi in ordine crescente i soli interi dispari in **X**,
- se **X** non contiene interi dispari, il programma deve stampare il messaggio

```
“Non esistono interi dispari!”
```