

La presente esercitazione presuppone conoscenza delle seguenti parti del linguaggio C: main, return, commenti, variabili, identificatori, tipi scalari, assegnazione, costanti letterali, printf (base), scanf (deprecata), operazioni aritmetiche, if...else, operatori relazionali, operatori Booleani, getchar, putchar, ciclo while, libreria ctype.h, const, #define, exit, programmazione a stati, gets, sscanf, stringhe, traduzione di algoritmi matematici.

Esercizio 1

Scrivere un programma in linguaggio C che riceva in input tre numeri interi sulla stessa riga e ne calcoli la somma. Se sono presenti meno di tre numeri sulla stessa riga, il programma deve segnalare errore e non svolgere alcun calcolo.

Esercizio 2

Scrivere un programma in linguaggio C che riceva in input sulla stessa riga fino ad un massimo di tre numeri interi e calcoli la media aritmetica dei numeri introdotti. Nel caso che non sia introdotto nessun numero, il programma deve terminare silenziosamente.

Esercizio 3

Scrivere un programma in linguaggio C per calcolare il valore della seguente espressione algebrica:

$$ax^3 + bx^2 + cx + d$$

Il programma riceve in input sulla prima riga i valori dei quattro coefficienti (a , b , c e d) e quindi sulla seconda riga il valore di x .

Esercizio 4

Scrivere un programma in linguaggio C per calcolare il valore della seguente espressione algebrica:

$$ax^3 + bx^2 + cx + d$$

Il programma riceve in input sulla prima riga i valori dei quattro coefficienti (a , b , c e d) e quindi sulle righe successive i valori di x (uno per riga) per i quali si desidera calcolare il valore dell'espressione. Nel caso una riga non contenga il valore di x in output dovrà comparire il testo "valore di X non specificato". Il programma dovrebbe quindi comportarsi come esemplificato nella seguente tabella:

<i>input</i>	→	<i>output</i>
1 1 1 2		
0		2
1		5
due		valore di X non specificato
2		16

Esercizio 5

Scrivere un programma in linguaggio C per trovare in quale ascissa (all'interno dell'intervallo fornito dall'utente) la seguente funzione si azzera:

$$y(x) = ax^3 + bx^2 + cx + d$$

Il programma riceve in input sulla prima riga i valori dei quattro coefficienti (a , b , c e d), quindi sulla seconda riga gli estremi dell'intervallo di ricerca $[L, H]$ ed infine sulla terza riga la precisione assoluta desiderata (ϵ). Il programma deve quindi trovare un valore $x_0 \in [L, H]$ tale che $|y(x_0)| \leq \epsilon$. Si suggerisce di procedere usando l'algoritmo di bisezione:

06AZN - Fondamenti di Informatica (GES, LOP, ORG)
Esercitazione di laboratorio n. 4 (3/11/09)

1. si consideri un intervallo $[L, H]$ ai cui estremi la funzione assume valori di segno opposto (se l'intervallo non gode di questa proprietà allora non si può applicare l'algoritmo);
2. si consideri il punto intermedio dell'intervallo, x_M ;
3. se $|y(x_M)| \leq \varepsilon$ l'algoritmo termina (x_M è il punto cercato), altrimenti si ricomincia dal passo 2 considerando come intervallo di ricerca quel sotto-intervallo, $[L, x_M]$ oppure $[x_M, H]$, ai cui estremi la funzione assume valori di segno opposto (se nessuno dei due sotto-intervalli gode di questa proprietà allora l'algoritmo termina senza essere riuscito a trovare uno zero della funzione).

Ad esempio se l'utente cercasse uno zero della funzione $y = x + 5$ nell'intervallo $[-100, 100]$ con la precisione di $1/1000$ allora dovrebbe introdurre in input i seguenti dati:

```
0 0 1 5
-100 100
0.001
```

e l'algoritmo inizierebbe eseguendo i seguenti passi:

- l'intervallo iniziale $[-100, 100]$ è valido poiché $y(-100) = -95$ ha segno opposto a $y(100) = 105$
- considero $x_M = 0$, quindi $y(0) = 5$ e $|y(0)| > 0.001$
- considero il sotto-intervallo $[-100, 0]$
- considero $x_M = -50$, quindi $y(-50) = -45$ e $|y(-50)| > 0.001$
- considero il sotto-intervallo $[-50, 0]$
- considero $x_M = -25$, quindi $y(-25) = -20$ e $|y(-25)| > 0.001$
- considero il sotto-intervallo $[-25, 0]$
- ...