

Esercitazione 3

Nota 1 Per esprimere la ripetizione di una serie di istruzioni, il C dispone di forme alternative alla forma `while`. Una di queste forme è esemplificata da questo frammento di codice:

```
for ( i = 0; i < N; i = i+1 ) {  
    /* Serie di istruzioni */  
}
```

che equivale a

```
i = 0;  
while ( i < N ) {  
    /* Serie di istruzioni */  
    i = i + 1;  
}
```

1 Scrivete una funzione che riceve come argomenti un numero reale x e un intero non negativo n , e restituisce il valore di x^n . Incorporate la funzione in un programma che legge x e n , invoca la funzione e stampa il risultato. L'istruzione di stampa non deve far parte della funzione. Verificate che il programma calcoli il risultato corretto quando x vale 0, 1 o -1, e quando n vale 0.

2 Scrivete un programma che dichiara un vettore di N elementi di tipo `int` e lo riempie leggendo i valori dallo *standard input*. Successivamente il programma determina la posizione del valore minimo nel vettore, e lo scambia con il valore nella prima posizione. Infine, il programma stampa i valori nel vettore.

3 Scrivete un programma che dichiara una matrice di dimensione $M \times N$ e la riempie di valori di tipo `double`, leggendoli dallo *standard input*. Successivamente, il programma calcola la somma di tutti i valori nella matrice, la somma dei valori della seconda colonna, la somma dei valori della terza riga. Definite M e N attraverso direttive `define`, in modo che $M \geq 3$ e $N \geq 2$.

4 Inventate un algoritmo per stabilire se un numero intero positivo è primo. Traducete l'algoritmo in una funzione, in C, con un parametro, e inseritela nel programma `primi.c`. Sperimentate il programma con diversi valori. Aggiungete una nuova funzione che riceve come parametro un numero intero, k , e stampa tutti i numeri primi minori o uguali a k , servendosi della prima funzione. Completate il programma facendo in modo che nella funzione `main` il programma chieda di inserire un numero intero n e richiami la seconda funzione.