
INTERMEDIO DI PROGRAMMAZIONE I

11 FEBBRAIO 2019

INDICAZIONI GENERALI

- Utilizzare il comando `ulimit -v 500000` per limitare l'utilizzo delle risorse al terminale su cui viene eseguito il comando ed evitare spiacevoli inconvenienti dovuti ad eccessive allocazioni di memoria.
 - Scaricare il file di ogni esercizio e riconsegnarlo senza modificarne il nome.
 - I file non consegnati o consegnati con errori di compilazione non verranno presi in considerazione.
 - Si possono utilizzare funzioni aggiuntive non presenti nei file modello e aggiungere linee di commento alle funzioni già implementate nel modello.
 - I file possono essere consegnati più volte. Per ogni esercizio, solo l'ultimo file consegnato sarà considerato valido.
 - I warning ottenuti compilando con l'opzione `-Wall` avranno un peso negativo sul voto finale.
-

COMPITO A

Esercizio 1 [11 punti] File **ESA_11022019_A_1.c**

Si completi il file **ESA_11022019_A_1.c** in modo che che il programma:

-) dichiari una matrice di nome **Matrix**, di dimensione $N \times M$ (dove N e M sono definiti a inizio programma come $N=2$, $M=3$), di numeri interi;
-) dichiari un vettore di decimali di nome **Filter** di dimensione M ;
-) inizializzi la matrice t.c. le righe pari (incluso riga 0) abbiano numeri random pari e le righe dispari abbiano numeri random dispari, sia pari che dispari compresi tra 0 e 10 (si ricorda che $2 \cdot x$ è sempre un numero pari e che $2 \cdot x + 1$ è sempre un numero dispari);
-) chieda all'utente l'inserimento dei M valori per **Filter**;
-) stampi **Matrix** e **Filter**. Un esempio con $N=3$, $M=2$, e 0.0, 1.0, 2.0 inseriti dall'utente per **Filter** è il seguente:

Matrix:

```
2  2  10
3  7  3
```

Filter:

```
0.0  1.0  2.0
```

-) implementi il prodotto matrice x vettore e stampi il risultato. Si ricorda che il risultato è dato da un vettore **Result[]** di dimensione N , ove ogni elemento c_i è calcolato come:

$$c_i \sum_{j=0}^{M-1} m_{i,j} * f_j = m_{i,1} * f_1 + m_{i,2} * f_2 + \dots + m_{i,M-1} * f_{M-1}$$

dove $m_{i,j}$ è l'elemento di **Matrix** riga i , colonna j mentre f_j è il j -esimo elemento del vettore **Filter**.

Ad esempio, il risultato del prodotto tra matrice e vettore dell'esempio sopra è il seguente:

Result:

```
22.0  13.0
```

Esercizio 2 [11 punti] File ESA_11022019_A_2.c

Un numero è detto “troncabile primo destro” se il numero stesso e tutti i numeri ottenuti rimuovendo successivamente la cifra più a destra sono primi. Scrivere nel file `ESA_11022019_A_2.c` il sottoprogramma `int rightprime(int val)` che ricevuto in ingresso un intero `val` sicuramente positivo restituisce 1 se il numero è troncabile primo destro, 0 altrimenti. Ad esempio, se al sottoprogramma viene passato in ingresso il numero 317 esso restituirà 1, infatti 317 è primo, 31 è primo e 3 è primo.

Esercizio 3 [11 punti] File ESA_11022019_A_3.c

I punti del piano cartesiano possono essere definiti utilizzando due vettori monodimensionali di uguale dimensione, il primo per memorizzare i valori delle ascisse e il secondo per i corrispondenti valori delle ordinate. Si completi il file `ESA_11022019_A_3.c` in modo tale che il seguente programma (già presente nello file stub):

```
int main(){
    double xs[] = {0, -1.1, 2, 0.9}; // Ascisse
    double ys[] = {0.5, 0, 1, 0}; // Ordinate
    int size = 4;
    double radius = 1;

    print(xs, ys, size);
    printf("Numero di punti dentro il cerchio di raggio %.2lf centrato nell'origine: %i\n",
           radius, count_in(xs, ys, size, radius));
    return 0;
}
```

produca il seguente risultato:

```
(0.00, 0.50)
(-1.10, 0.00)
(2.00, 1.00)
(0.90, 0.00)
Numero di punti dentro il cerchio di raggio 1.00 centrato nell'origine: 2
```