
INTERMEDIO DI PROGRAMMAZIONE I

3 FEBBRAIO 2021

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_03022021_A_1.c`

Completare **tutte le funzioni dichiarate ma non definite** nel programma `ESA_03022021_A_1.c` per un vettore di interi che utilizza il valore intero "speciale" `END` come terminatore.

In particolare, la funzione `main()` fornita produce il seguente output:

Somma: 47

Somma: 30

Esercizio 2 [11 punti] File `ESA_03022021_A_2.c`

Si completi il file `ESA_03022021_A_2.c` definendo il sottoprogramma `void codifica(char [], int, int)` che ricevuti in ingresso un array di `char`, la sua dimensione e un intero k sicuramente non negativo, sostituisce nell'array ogni **lettera dell'alfabeto minuscola** con la lettera successiva (la lettera 'z' verrà sostituita con 'a'), se la lunghezza del numero k è un valore pari, con la lettera precedente (la lettera 'a' verrà sostituita con 'z'), se la lunghezza di k è un valore dispari. Ad esempio richiamando il sottoprogramma e passando come parametri l'array

```
a[]={ '1', 'a', 'Z', 'd', '9', '*', 'z', 'z', 'c', 'a' }
```

di lunghezza 10 e il valore $k = 34$ (di lunghezza 2), il sottoprogramma cambierà l'array nel seguente modo

```
{ '1', 'b', 'Z', 'e', '9', '*', 'a', 'a', 'd', 'b' }
```

Se $k = 135$ (di lunghezza 3), il sottoprogramma cambierà l'array nel seguente modo

```
{ '1', 'z', 'Z', 'c', '9', '*', 'y', 'y', 'b', 'z' }
```

Si possono definire sottoprogrammi ulteriori, se ritenuto opportuno.

Esercizio 3 [11 punti] File `ESA_03022021_A_3.c`

Si completi il file `ESA_03022021_A_3.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 due vettore di interi di nome `SumR` e `SumC` di dimensione N e M rispettivamente;
-) 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 $2*x$ è sempre un numero pari e che $2*x+1$ è sempre un numero dispari);

-) implementi la somma per righe e per colonne salvando i risultati in SumR (SumR[0] conterrà la somma degli interi di Matrix riga 0, SumR[1] la somma di riga 1 etc.) e SumC (SumC[0] conterrà la somma degli interi di Matrix colonna 0, SumC[1] la somma di colonna 1, etc.);

-) stampi Matrix e Sum.

Un esempio con N=3, M=2, e il seguente:

Matrix:

2	2	10
3	7	3

SumR:

14	13
----	----

SumC:

5	9	13
---	---	----