
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 C

Esercizio 1 [11 punti] File `ESA_03022021_C_1.c`

Completare **tutte le funzioni dichiarate ma non definite** nel programma `ESA_03022021_C_1.c` per un particolare tipo di vettore di interi, nel quale il primo valore del vettore serve solo per indicare la lunghezza del vettore stesso e non deve essere considerato tra i suoi elementi "validi".

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

```
15 9 22 32 12
I multipli di 3 sono: 3
0 9 0 32 12
I multipli di 3 sono: 2
```

Esercizio 2 [11 punti] File `ESA_03022021_C_2.c`

Si completi il file `ESA_03022021_C_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 positivo, calcola la somma di tutte le cifre di k , ottenendo così un valore intero tot e sostituisce i primi tot caratteri dell'array con il carattere successivo secondo la codifica ASCII (non serve conoscere il valore intero associato ad ogni carattere per risolvere l'esercizio) se si tratta di una lettera dell'alfabeto maiuscola o minuscola, con il carattere precedente secondo la codifica ASCII in tutti gli altri casi. Ad esempio richiamando il sottoprogramma e passando come parametri l'array

```
a[]={'1','a','Z','d','8','1','y','e','c','a'}
```

di lunghezza 10 e il valore $k = 34$ (la somma delle cifre vale $3 + 4 = 7$), il sottoprogramma cambierà l'array **solo** nelle prime 7 posizioni nel seguente modo

```
{'0','b','[','e','7','0','z','e','c','a'}
```

Esercizio 3 [11 punti] File `ESA_03022021_C_3.c`

Si completi il file `ESA_03022021_C_3.c` in modo che il programma:

-) dichiari una matrice costante di nome `A`, di dimensione $N \times M$ (dove N e M sono definiti a inizio programma come $N=2$ e $M=3$), di numeri interi;
-) dichiari una matrice di interi `B` di dimensione $N \times M$;
-) inizializzi `A` come segue:

A:

```
18  -1  -21
26  33  -49
```

-) chieda all'utente l'inserimento di tutti i valori interi per B;
-) calcoli la somma $A+B$ e la salvi in una matrice Sum;
-) stampi la matrice Sum, il suo valore massimo e gli indici del valore massimo. Ad esempio, con i seguenti valori inseriti dall'utente: 1, 1, 1, 2, 2, 2 il programma stamperà:

Sum:

```
[ 19] [  0] [-20]
[ 28] [ 35] [-47]
```

Il valore massimo è 35 e si trova in Sum[1][1]