

Esame 20260108

Esercizio 3

(1) Esercizio 3 v1

ESSAY marked out of 10 penalty 0 File picker

Dato uno stack di interi, implementare la funzione `calcola` che calcola e ritorna la somma delle cifre di tutti gli interi presenti nello stack. Alla fine della computazione lo stack deve contenere gli stessi elementi che aveva all'inizio, nello stesso ordine.

Il `main` del programma è già implementato e non deve essere modificato, e chiama la funzione `calcola` (**da definire**) che prende come argomento uno stack di interi (prestare attenzione a come passare il parametro) e ritorna un intero che rappresenta la somma delle cifre di tutti gli interi presenti nello stack.

La funzione `calcola` **deve essere ricorsiva** e **NON deve contenere iteratori** espliciti (`for`, `while`, `do-while`). La funzione `calcola` può ovviamente contenere codice sequenziale o condizionale. Sono consentite (se ritenute necessarie) chiamate a funzioni ricorsive ausiliarie che a loro volta **non contengano iterazioni esplicite** (`for`, `while`, `do-while`).

Il file `esercizio3.cpp` contiene tutto quanto necessario tranne la dichiarazione e la definizione della procedura `calcola`. Insieme anche ai file `stack.h` e `stack.cpp` che implementano una pila di interi (già testata e funzionante).

Di seguito è riportato un esempio di esecuzione del programma.

```
computer > ./a.out
Contenuto della pila: 34 23 12
Risultato calcola: 15
Contenuto della pila: 34 23 12
Stack vuoto
Risultato calcola: 0
Stack vuoto
```

Note:

- Scaricare i file `esercizio3.cpp`, modificare il solo file `esercizio3.cpp` per inserire il codice necessario per rispondere a questo esercizio. **Caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `cstdlib`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).
- Si ricorda che la soluzione deve essere implementata in C++ **NON usando altri elementi della C++ standard template library tranne le funzioni definite nelle librerie concesse, anche se il file compila senza cambiare gli header!**

[esercizio3.cpp](#)

[stack.cpp](#)

[stack.h](#)

Information for graders:

(2) Esercizio 3 v2

ESSAY

marked out of 10

penalty 0

File picker

Dato uno stack di interi, implementare la funzione `calcola` che calcola e ritorna il prodotto delle cifre di tutti gli interi presenti nello stack. Alla fine della computazione lo stack deve contenere gli stessi elementi che aveva all'inizio, nello stesso ordine.

Il main del programma é già implementato e non deve essere modificato, e chiama la funzione `calcola` (**da definire**) che prende come argomento uno stack di interi (prestare attenzione a come passare il parametro) e ritorna un intero che rappresenta il prodotto delle cifre di tutti gli interi presenti nello stack.

La funzione `calcola` **deve essere ricorsiva** e **NON deve contenere iteratori** espliciti (`for`, `while`, `do-while`). La funzione `calcola` può ovviamente contenere codice sequenziale o condizionale. Sono consentite (se ritenute necessarie) chiamate a funzioni ricorsive ausiliarie che a loro volta **non contengano iterazioni esplicite** (`for`, `while`, `do-while`).

Il file `esercizio3.cpp` contiene tutto quanto necessario tranne la dichiarazione e la definizione della procedura `calcola`. Insieme anche ai file `stack.h` e `stack.cpp` che implementano una pila di interi (già testata e funzionante).

Di seguito é riportato un esempio di esecuzione del programma.

```
computer > ./a.out
Contenuto della pila: 34 23 12
Risultato calcola: 144
Contenuto della pila: 34 23 12
Stack vuoto
Risultato calcola: 0
Stack vuoto
```

Note:

- Scaricare i file `esercizio3.cpp`, modificare il solo file `esercizio3.cpp` per inserire il codice necessario per rispondere a questo esercizio. **Caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream` e `cstdlib`.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta **NON** deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).
- Si ricorda che la soluzione deve essere implementata in C++ **NON usando altri elementi della C++ standard template library** tranne le funzioni definite nelle librerie concesse, anche se il file compila senza cambiare gli header!

`esercizio3.cpp`

`stack.cpp`

`stack.h`

Information for graders:

Total of marks: 20