

Esame 20250909

Esercizio 2

(1) Esercizio 2 v1

Nel sollevamento pesi, i dischi vengono impilati verticalmente su un bilanciere. Ogni peso ha una massa m in kg, e la sua energia potenziale gravitazionale è data dalla formula $E = m \cdot g \cdot h$, dove g è l'accelerazione di gravità (assunta pari a 9.81 m/s^2), e h è l'altezza del bilanciere da terra. Si assuma che ogni bilanciere abbia una altezza pari a 1.5m. Si vuole scrivere una funzione ricorsiva `calcola` che, dato uno stack di pesi, calcoli e stampi a video l'energia potenziale totale dei pesi impilati sul bilanciere, lasciando inalterato il contenuto dello stack dei pesi. La funzione `calcola` prende come argomento lo stack dei pesi, e deve essere implementata in C++ senza l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`. 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. Ad esempio, se lo stack contiene i pesi 10, 15 e 20 kg, l'output deve essere: $10*9.81*0 + 15*9.81*1.5 + 20*9.81*3 = 809.32 \text{ J}$. Se lo stack contiene i pesi 20, 15 e 10 kg, l'output deve essere: $20*9.81*0 + 15*9.81*1.5 + 10*9.81*3 = 515.025 \text{ J}$. Se lo stack contiene solo il peso 10 kg, l'output deve essere $10*9.81*0 = 0 \text{ J}$.

Il `main` del programma è già implementato e non deve essere modificato, e chiama la funzione `calcola` (**da definire**) che prende gli argomenti sopra specificati. La funzione `calcola` ritorna un `double` che rappresenta l'energia potenziale dell'intero stack di pesi.

La funzione `calcola` **deve essere ricorsiva** e **NON deve contenere iteratori esplicativi** (`for`, `while`, `do-while`). La funzione `cacola` 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 `esercizio2.cpp` contiene tutto quanto necessario tranne la dichiarazione e la definizione della procedura `calcola`.

Di seguito è riportato un esempio di esecuzione del programma.

```
computer > ./a.out
Contenuto della pila: 20 15 10
Contenuto della pila: 20 15 10
Energia calcolata: 515.025
-----
Contenuto della pila: 10
Contenuto della pila: 10
Energia calcolata: 0
-----
Stack vuoto
Stack vuoto
Energia calcolata: 0
```

Note:

- Scaricare i file `esercizio2.cpp`, `stack.cpp` e `stack.h`, modificare il solo file `esercizio2.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.
- Si può assumere che la stringa di input contenga solo caratteri numerici e spazi.
- 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`, `fstream`, `cstdlib`, `cstdio`.

- 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 string e altri elementi della C++ standard template library, anche se il file compila senza cambiare gli header!**

esercizio2.cpp

stack.h

stack.cpp

Information for graders:

Total of marks: 10