

Esame 20230830

Esercizio 2

(1) Esercizio 2 v1

ESSAY marked out of 10 penalty 0 File picker

Un *numero poligonale* é un numero intero figurato che può essere disposto a raffigurare un poligono regolare. In letteratura esistono diversi numeri poligonali: e.g., triangolari, quadrati, pentagonali, esagonali.

Scrivere nel file `esercizio2.cpp` la dichiarazione e la definizione della **funzione ricorsiva** `crea_lista` che, dato un intero positivo N crea e ritorna una lista concatenata contenente i primi N numeri triangolari, cominciando dallo 0.

L' i -esimo numero triangolare si ottiene con la formula $T(i) = \frac{i(i+1)}{2}$.

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

La funzione `crea_lista` è inserita in un `main` che legge dalla command line un intero N , chiama la funzione stessa e stampa la lista costruita, e poi la dealloca.

Di seguito sono riportati alcuni esempi di esecuzione con alcuni parametri.

```
marco > ./a.out 8
N = 8
The list of taylor terms is: 28 21 15 10 6 3 1 0
marco > ./a.out 12
N = 12
The list of taylor terms is: 66 55 45 36 28 21 15 10 6 3 1 0
```

Note:

- Scaricare il file `esercizio2.cpp`, modificarlo per inserire la dichiarazione e la definizione della funzione `crea_lista`, e **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`, `cstdlib`, `ctime`, `io manip`, e `cmath`.
- 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.

`esercizio2.cpp`

Information for graders:

(2) Esercizio 2 v2

ESSAY marked out of 10 penalty 0 File picker

Un *numero poligonale* é un numero intero figurato che può essere disposto a raffigurare un poligono regolare. In letteratura esistono diversi numeri poligonali: e.g., triangolari, quadrati, pentagonali, esagonali.

Scrivere nel file `esercizio2.cpp` la dichiarazione e la definizione della **funzione ricorsiva** `crea_lista` che, dato un intero positivo N crea e ritorna una lista concatenata contenente i primi N numeri esagonali, cominciando dallo 0.

L' i -esimo numero esagonale si ottiene con la formula $E(i) = i(2i - 1)$.

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

La funzione `crea_lista` è inserita in un `main` che legge dalla command line un intero N , chiama la funzione stessa e stampa la lista costruita, e poi la dealloca.

Di seguito sono riportati alcuni esempi di esecuzione con alcuni parametri.

```
marco > ./a.out 8
N = 8
The list of taylor terms is: 91 66 45 28 15 6 1 0
marco > ./a.out 12
N = 12
The list of taylor terms is: 231 190 153 120 91 66 45 28 15 6 1 0
```

Note:

- Scaricare il file `esercizio2.cpp`, modificarlo per inserire la dichiarazione e la definizione della funzione `crea_lista`, e **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`, `cstdlib`, `ctime`, `io manip`, e `cmath`.
- 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.

`esercizio2.cpp`

Information for graders:

(3) Esercizio 2 v3

ESSAY marked out of 10 penalty 0 File picker

Un *numero poligonale* é un numero intero figurato che può essere disposto a raffigurare un poligono regolare. In letteratura esistono diversi numeri poligonal: e.g., triangolari, quadrati, pentagonali, esagonali.

Scrivere nel file `esercizio2.cpp` la dichiarazione e la definizione della **funzione ricorsiva** `crea_lista` che, dato un intero positivo N crea e ritorna una lista concatenata contenente i primi N numeri pentagonali, cominciando dallo 0.

L' i -esimo numero pentagonale si ottiene con la formula $E(i) = \frac{i(3i-1)}{2}$.

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

La funzione `crea_lista` è inserita in un `main` che legge dalla command line un intero N , chiama la funzione stessa e stampa la lista costruita, e poi la dealloca.

Di seguito sono riportati alcuni esempi di esecuzione con alcuni parametri.

```
marco > ./a.out 8
N = 8
The list of taylor terms is: 70 51 35 22 12 5 1 0
marco > ./a.out 12
N = 12
The list of taylor terms is: 176 145 117 92 70 51 35 22 12 5 1 0
```

Note:

- Scaricare il file `esercizio2.cpp`, modificarlo per inserire la dichiarazione e la definizione della funzione `crea_lista`, e **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`, `cstdlib`, `ctime`, `iomanip`, e `cmath`.
- 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.

`esercizio2.cpp`

Information for graders:

Total of marks: 30