Esercizi 15-10-2024

1

Scrivere una funzione ricorsiva che calcoli il fattoriale di un qualsiasi numero intero positivo. Riscrivere poi la sua versione iterativa (quale delle due è più efficiente?).

Bonus:
Usare la libreria `chrono` (`#include <chrono>`) per misurare quanto dura l'esecuzione di ogni funzione:

```
chrono::time_point<chrono::steady_clock> start, end;
chrono::duration<double> time_span;

start = chrono::steady_clock::now();
// chiamata alla funzione
end = chrono::steady_clock::now();
time_span = chrono::duration_cast<chrono::duration<double>>(end - start);
cout << "Tempo trascorso: " << time_span.count() << " secondi" << endl;
```

Prendendo il tempo prima e dopo la funzione che si vuole misurare (`start` e `end`) è possibile sottrarli e ottenere la durata.

2

Creare una funzione che riceve un numero e restituisce la somma delle cifre del numero solo quando questa è minore di 10. Deve altrimenti restituire il risultato della funzione applicata alla somma delle cifre del numero.
Ad esempio:

1. riceve 15, esegue `f(15) = 1 + 5 = 6`, restituisce 6
2. riceve 392, esegue `f(392) = 3 + 9 + 2 = 14 > 10`, `f(14) = 1 + 4 = 5`, restituisce 5

(*suggerimento: usare una funzione d'appoggio ricorsiva che controlli la somma*)

3

Data una media iniziale e un numero di crediti totali acquisiti, far inserire all'utente il voto e i crediti di un nuovo esame e scrivere una funzione per ricalcolare la nuova media pesata.

Bonus: Scrivere una funzione con passaggio per valore, una con passaggio per riferimento e una con passaggio per puntatore. Provare ad eseguire ognuna delle funzioni e capire cosa succede.

4

Dati in input tre interi positivi in tre variabili (n1, n2, n3), scrivere una funzione che riordini e stampi i numeri in ordine crescente.

Bonus: Scrivere una funzione con passaggio per valore, una con passaggio per riferimento e una con passaggio per puntatore. Provare ad eseguire ognuna delle funzioni e capire cosa succede.

5

Dato in input un numero `n`, in maniera ricorsiva chiedere all'utente di inserire una lettera per `n` volte e, sapendo che le lettere in maiuscolo valgono 10 mentre quelle in minuscolo valgono 5, calcolare e stampare a video la somma totale. È possibile usare la funzione `islower(char)` contenuta nella libreria `cctype` per controllare che il carattere in input sia minuscolo.

6

Dato un array inizializzato in modo casuale, ordinarlo usando *StalinSort*, descritto nella seguente procedura:

- Confrontare i due valori iniziali nell'array;
- Se il primo è maggiore del secondo, eliminalo (settando a 0).
- Ripeti per tutti i valori

7

Data la seguente coppia di array di dimensione 15:

```
// valori del primo array
{88, 76, 21, 72, -2, 24, 63, 101, 35, 100, 105, 81, 5, 33, 66}
// valori del secondo array
{45, 76, 100, 34, 10, 5, 81, 13, 48, 78, 34, 39, 89, 35, 27}
```

Scrivi un programma che esegua le seguenti operazioni:

- Somma ogni elemento del primo array con l'elemento corrispondente del secondo array, considerando quest'ultimo "girato"

  ```
  Il primo elemento del primo array va sommato con l'ultimo del secondo array.
  Il secondo elemento del primo array va sommato con il penultimo del secondo arr
  Continua questo processo fino a esaurire tutti gli elementi di entrambi gli arr
  ```

- Stampa il risultato di ogni somma in sequenza.

Puoi assumere che entrambi gli array abbiano la stessa lunghezza.

Potresti defire con `#define` una costante?

8

FoldLeft è un'operazione molto usata nella programmazione funzionale. Solitamente prende come parametro un array, un valore iniziale e lo step a cui siamo nell'array. La funzione viene applicata a tutti gli elementi dell'array, partendo dal primo, e il risultato viene passato come parametro alla funzione per il prossimo elemento.

Ad esempio, se abbiamo un array di interi e vogliamo calcolare la somma di tutti gli elementi, possiamo usare la funzione `foldLeft` in questo modo (NB: alcuni step sono mancanti nella funzione seguente, è necessario aggiungerli per farla funzionare):

```
int foldLeft(array, init) {
    val = get_valore_from_array(array);
    return foldLeft(array, init + val);
};
```

Scrivere una funzione `foldLeft` che prenda in input un array, un valore iniziale e l'indice corrente, e restituisca il massimo valore dell'array.

Hint: è necessario aggiungere un caso base per la ricorsione.