

Esercizi - C++: Gli Array e La memoria Dinamica

1. Ordinare un array

Scrivere una funzione *template* che, dato un array di valori, lo ordini.

2. I numeri primi

Scrivere una funzione che calcoli i primi n numeri primi e li restituisca.

3. La media e la varianza

Scrivere una funzione che calcoli la media di un insieme di valori contenuti in un array.

Scrivere una funzione che calcoli la varianza di un insieme di valori contenuti in un array.

4. Il cambiamonete

Scrivere una funzione `cambia` che date:

1. un array di possibili tagli di monete di una valuta es. 0.05 euro, 0.10 euro , 0.20 euro, e 1 euro
2. il numero di monete presenti nel cambiamonete per ogni taglio
3. un valore da cambiare es. 3.45 euro

Restituisca, dove possibile, un array di lunghezza minima di monete presenti nel cambiamonete la cui somma delle valute equivalga al valore che vogliamo cambiare.

Testare la funzione `cambia` con assumendo che si voglia cambiare 1.17 euro che il cambia monete contenga 30 monete da 0.01 euro, 4 monete 0.29 euro e 2 monete da 1 euro.

5. Fibonacci e *dynamic programming*

Il *dynamic programming* è una tecnica che consente di accelerare la valutazione di funzioni ricorsive i cui passi ricorsivi siano fortemente ridondanti.

Per esempio, la valutazione del n -esimo numero di Fibonacci, i.e., $Fib(n)$, richiede la computazione di $Fib(n - 1)$ e $Fib(n - 2)$, ma anche $Fib(n - 1)$ richiede la computazione di $Fib(n - 2)$.

Per evitare la di valutare due volte la stessa istanza del problema, possiamo usare uno spazio di memoria ausiliario per *memoizzare* i risultati di ciascuna istanza. Nel caso in cui una specifica istanza sia già stata valutata, la soluzione verrà recuperata dalla memoria anziché ricalcolata.

Modifica la versione ricorsiva della funzione che calcola l' n -esimo numero di Fibonacci utilizzando un array ausiliario per memoizzare $Fib(i)$ per tutti gli $i \in [0, n - 1]$.