

Esercizio 2

(1) Esercizio 2 v1

ESSAY marked out of 10 penalty 0 File picker

Data una sequenza di caratteri (rappresentata con un array di dimensione D) che rappresenta ad esempio una lista di azioni che un agente robotico deve eseguire nell'ordine specificato nella sequenza, e due indici i e j con $0 \leq i \leq j < D$, calcolare i) quante volte l'azione nella posizione j **occorre** nelle posizioni k tali che $0 \leq i \leq k < j$, e ii) tutti gli indici k dell'array.

A tale scopo si scriva nel file `esercizio2.cpp` la dichiarazione e la definizione della **procedura ricorsiva** `conta` che prende come argomenti: i) un array A di `char`; ii) un intero D che rappresenta la dimensione dell'array A ; iii) un intero i ; iv) un intero j ; e v) un array di interi res passato per *riferimento*.

La procedura dovrà **allocare dinamicamente** l'array res . L'array res dovrà avere una **dimensione tale da contenere la lista di tutti e soli gli indici che soddisfano la condizione specificata più uno (per contenere nella posizione 0 il numero delle occorrenze che soddisfano la condizione)**. La dimensione di res deve essere la **minima possibile** (ovvero non deve consentire di aggiungere altri interi oltre a quelli strettamente necessari).

La procedura `conta` **deve essere ricorsiva** e **NON deve contenere iteratori** espliciti (`for`, `while`, `do-while`). La procedura `conta` 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`).

La procedura `conta` è inserita in un `main` che crea una sequenza random, ne stampa i valori, genera due indici i, j , chiama la procedura `conta`, stampa il risultato della chiamata alla procedura, e dealloca l'array allocato dalla procedura `conta`.

Di seguito è riportato un esempio di esecuzione.

```
marco > a.out
Seed = 1697033220
A[] = PNKSSWOMLEEIUIQCQFWGHAMVMOEIVTOMOILUUBXQGIMSFVQKLWKLCAQDAGXOSFWVRKPJCNJ
i = 4, j = 52
A[4] = S A[52] = W
There are 3 occurrences of the searched pattern between indexes 4 and 52 (with A[4] included and A[52] excluded)
They are at indexes: 5 8 20
```

Note:

- Scaricare il file `esercizio2.cpp`, modificarlo per inserire la dichiarazione e la definizione della funzione `conta`, 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`, `ioomanip`, 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 (pena annullamento dell'esercizio).

esercizio2.cpp

Information for graders:

(2) Esercizio 2 v2

ESSAY marked out of 10 penalty 0 File picker

Data una sequenza di caratteri (rappresentata con un array di dimensione D) che rappresenta ad esempio una lista di azioni che un agente robotico deve eseguire nell'ordine specificato nella sequenza, e due indici i e j con $0 \leq i \leq j < D$, calcolare i) quante volte l'azione nella posizione j **non occorre** nelle posizioni k tali che $0 \leq i \leq k < j$, e ii) tutti gli indici k dell'array.

A tale scopo si scriva nel file `esercizio2.cpp` la dichiarazione e la definizione della **procedura ricorsiva** `conta` che prende come argomenti: i) un array A di `char`; ii) un intero D che rappresenta la dimensione dell'array A ; iii) un intero i ; iv) un intero j ; e v) un array di interi res passato per *riferimento*.

La procedura dovrà **allocare dinamicamente** l'array res . L'array res dovrà avere una **dimensione tale da contenere la lista di tutti e soli gli indici che soddisfano la condizione specificata più uno (per contenere nella posizione 0 il numero delle occorrenze che soddisfano la condizione)**. La dimensione di res deve essere la **minima possibile** (ovvero non deve consentire di aggiungere altri interi oltre a quelli strettamente necessari).

La procedura `conta` **deve essere ricorsiva** e **NON deve contenere iteratori** espliciti (`for`, `while`, `do-while`). La procedura `conta` 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`).

La procedura `conta` è inserita in un `main` che crea una sequenza random, ne stampa i valori, genera due indici i, j , chiama la procedura `conta`, stampa il risultato della chiamata alla procedura, e dealloca l'array allocato dalla procedura `conta`.

Di seguito è riportato un esempio di esecuzione.

```
marco > ./a.out
Seed = 1697033220
A[] = PNXSSWOMWLEEUUIQQQFWGHAMVMOEIVTOMOILUUBXQGIMSIFVQKLWKLCAQDAGXOSFWVRKPJCNJ
i = 4, j = 52
A[4] = S A[52] = W
There are 45 occurrences of the searched pattern between indexes 4 and 52 (with A[4] included and A[52] excluded)
They are at indexes: 4 6 7 9 10 11 12 13 14 15 16 17 18 19 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
```

Note:

- Scaricare il file `esercizio2.cpp`, modificarlo per inserire la dichiarazione e la definizione della funzione `conta`, 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 (pena annullamento dell'esercizio).

[esercizio2.cpp](#)

Information for graders:

Total of marks: 20