

Temă pentru acasă. Prelucrări de stringuri

Completați definițiile următoarelor funcții corespunzător cerințelor precizate. Se presupune că este definită o constantă globală

```
const int dim=1024;
```

care precizează dimensiunea maximă a tablourilor utilizate. Atenție, se vor utiliza numai stringuri în stil C (*null-terminated strings*).

- 1) `int lung(char text[dim]){...}`
returnează lungimea stringului `text` (numărul de caractere fără terminatorul de string).
- 2) `int de_cate_ori(char text[dim], char lit){...}`
numără de câte ori apare caracterul `lit` în stringul `text`.
- 3) `int clasifica(char lit){...}`
returnează 0 dacă `lit` este o literă mică, 1 dacă este o literă mare, 2 dacă este o cifră, iar în rest returnează -1.
- 4) `int evaluare(char lit){...}`
returnează, dacă `lit` este o cifră, valoarea numerică a acestei cifre, altfel returnează -1.
- 5) `void schimba_cazul(char text[dim], char star){...}`
modifică stringul `text` astfel: literele mari trec în litere mici, literele mici în litere mari, iar celelalte caractere sunt înlocuite cu `star`.
- 6) `int are_dubluri(char text[dim]){...}`
returnează 1 dacă stringul `text` conține două caractere identice, altfel returnează 0.
- 7) `void afiseaza(){...}`
citește un string de la tastatură și îl afișează pe monitor cu literele în ordine inversă.
- 8) `void copie(char dest[dim], char sursa[dim]){...}`
copie prin suprascriere stringul `sursa` în stringul `dest`.
- 9) `void concat(char dest[dim], char sursa[dim]){...}`
adaugă stringul `sursa` la sfârșitul stringului `dest`.
- 10) `void cifreaza(char inainte[dim], char inapoi[dim]){...}`
modifică stringul `inainte` înlocuind fiecare literă cu litera următoare din alfabet (`a` trece în `b`, ..., `y` trece în `z`, `z` trece în `a`). Caracterele care nu sunt litere sunt lăsate pe loc. Stringul `inapoi` este modificat în sens invers (`b` trece în `a`, etc.).

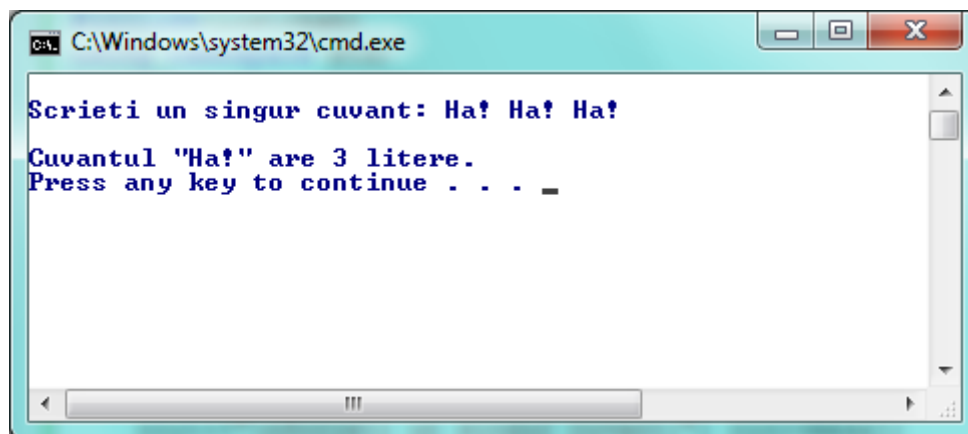
- 11) void elimina(char text[dim], char lit){...}
elimină din stringul text orice apariție a caracterului lit.
- 12) void dubleaza(char text[dim], char lit){...}
dublează în stringul text orice apariție a caracterului lit.

Exemplu de rezolvare:

```
#include<iostream>
using namespace std;
const int dim = 1024;
int lung(char text[dim]){
    int i;
    for(i=0; i<dim; i++)
        if(text[i]=='\0') break;
    return i;
}

int main(){
    char mesaj[dim];
    cout<<"\nScrieti un singur cuvant:"; cin>>mesaj;
    cout<<"\nCuvantul \""<<mesaj<<"\" are "<<lung(mesaj)<<" litere."<<endl;
    return 0;
}
```

Exemplu de rulare:



```
C:\Windows\system32\cmd.exe

Scrieti un singur cuvant: Ha! Ha! Ha!
Cuvantul "Ha!" are 3 litere.
Press any key to continue . . . _
```