

Teme și aplicații

Formatori:

Tutor: [Stângaciu Valentin](#)  

Tutor: [Belu Claudiu-Marcel](#)  

+3



Data de începere a cursului:

 25.09.2023

 [Utilizatori înscrși](#)

 [Calendar](#)

 [Note](#)

 [Cursurile mele](#) [S1-L-AC-CTIRO1-PC](#) [Laborator 12: Alocare dinamică](#) [Teme și aplicații](#)

Teme și aplicații

1. Să se scrie o funcție care primește două șiruri de caractere și le concatenează cu un spațiu între ele, folosind un bloc de memorie alocat dinamic pentru rezultat. Funcția va returna acest bloc de memorie. Să se verifice funcția cu două șiruri de caractere introduse de la tastatură, fiecare de maxim 100 de caractere.
2. Se citesc numere până la întâlnirea numărului 0. Să se afișeze aceste numere în ordine inversă. Programul va folosi doar minimul necesar de memorie
3. Jocul fazan: se citesc de la tastatură cuvinte separate prin spații albe (spațiu, TAB sau ENTER), până la întâlnirea cuvântului vid. Pornind de la cuvintele citite, se construiește un șir de caractere după următoarele reguli:
 - Primul cuvânt citit se adaugă la șirul de caractere
 - Fiecare din următoarele cuvinte citite se adaugă la șirul de caractere dacă ultimele două litere ale șirului coincid cu primele două litere ale cuvântului (nu se face distincție între literele mici și cele mari);Cuvintele adăugate la șirul de caractere sunt despărțite de caracterul '-'. Să se afișeze șirul astfel construit. Programul va folosi alocare dinamică astfel încât spațiul de memorie utilizat să fie minim.
Spre exemplu, pentru intrarea:

Fazan Antic Noe icoana Egipt naftalina nimic Narcisa trei altceva santier iarba Caine Pisica erudit

se afișează:

Fazan-Antic-icoana-naftalina-Narcisa-santier-erudit

4. Sa se scrie o functie cu urmatorul antent

```
char *string_replace_dynamic(const char *where, const char *what, const char *replace)
```

Funcția înlocuiește fiecare apariție a string-ului what în string-ul where cu string-ul replace. Funcția va returna noul string obținut, alocat dinamic, și NULL în caz de eroare sau în cazul în care string-ul what nu apare în string-ul where.

```
char s1[] = "Acesta este un string si un string este terminat cu 0x00";  
char s2[] = "string";  
char s3[] = "sir de caractere";
```

```
char *p = string_replace(s1, s2, s3);
```

p devine un string alocat dinamic cu urmatorul continut: "Acesta este un sir de caractere si un sir de caractere este terminat cu 0x00";
s1, s2, s3 - raman neschimbate
r = 2

5. Se va citi de la intrarea standard un numar neconscut de valor intregi nenule pozitive pana la EOF. Se vor stoca aceste elemente într-un tablou dinamic. Programul va afisa la iesirea standard tabloul sortat descrescator.
6. Să se scrie o funcție care primește ca argument un string și returnează un alt string alocat dinamic în care fiecare caracter din string-ul inițial va fi înlocuit cu valoarea în hexazecimal a caracterului conform tabelii ASCII în format text pe strict două cifre hexazecimale fără prefixul 0x. Antetul funcției poate fi următorul

```
char *text2hex(const char *string);
```

Exemplu: string = "abc" -> funcția va returna un string de forma "616263"

7. Să se scrie o funcție care primește ca argument un string ce conține în format text valori în hexazecimal pe două cifre hexazecimale fără prefixul 0x. Aceste valori reprezintă caractere conform tabelii ASCII. Funcția va decodifica textul și îl va returna într-un string alocat dinamic. Antetul funcției poate fi următorul:

```
char *hex2text(const char *hexstring);
```

Exemplu: hexstring = "616263" -> funcția va returna string-ul "abc"

Această problemă reprezintă inversa funcției dezvoltate la problema anterioară.

Pentru testare se pot folosi datele din fisierul ce se poate descarca prin comanda urmatoare

```
wget http://staff.cs.upt.ro/~valy/pt/hexlines.txt
```

Se recomanda ca citirea datelor sa se faca prin redirectarea intrarii standard.

8. Să se scrie o funcție ca primește ca parametru o propoziție și returnează un string alocat dinamic ce reprezintă propoziția citită translatată în "limba păsărească". Translatarea în "păsărească" constă în faptul că fiecare vocală întâlnită se dublează punându-se și caracterul 'p' înaintea vocalei dublate. Practic vocalele se transformă astfel:

a -> apa
e -> epe
i -> ipi
o -> opo
u -> upu

Exemplu: ana are mere și apa-> **apanapa aparepe meperepe sipi apapapa.**

◀ Considerații teoretice

Sari la...

Test grilă 4 ►

✉ Contactați serviciul de asistență

Sunteți conectat în calitate de
S1-L-AC-CTIRO1-PC

Meniul meu

Profil

Preferinte

Calendar

 ZOOM

Română (ro)

English (en)

Română (ro)

Rezumatul păstrării datelor

Politici utilizare site