***מחרוזות***

void str\_rev(char \*str)//כתיבת הפונקציה strrev

{

int i = 0, j = 0;

char temp;

while (str[i] != NULL)

{

j++;

i++;

}

for (i = 0, j = j - 1; i<j; i++, j--)

{

temp = str[i];

str[i] = str[j];

str[j] = temp;

}

}

find(char name[N], char chose[N])/\*כמה פעמים מופיע המילה המשפט \*/

{

{

int i = 0, j = 0, count = 0, size, round = 0;

size = strlen(chose);

do

{

if (name[i] == chose[j])

{

count++;

j++;

i++;

}

else

{

i++;

}

if (count == size)

{

round++;

count = 0;

j = 0;

}

} while (name[i] != '\0');

printf("How many times is the word |%s| displayed:%d\n",chose, round);

}

}

int strlen(char str1[])/\* כתיבה של הפונקציה strlen \*/

{

int i = 0; /\* index \*/

/\* strlen:check str1 \*/

i = 0;

while (str1[i] != NULL)

{

if (str1[i + 1] == NULL)

{

return i; /\* include null \*/

break;

}

i++;

}

}

int strcmp(char \*str1, char \*str2)/\* כתיבה של הפונקציה strcmp \*/

{

int i = 0, j = 0;

while (str1[i] != NULL&&str2[j] != NULL)

{

if (str1[i] > str2[j])

{

return 1;

}

if (str1[i] < str2[j])

{

return -1;

}

i++;

j++;

}

if (str1[i] != 0)

{

return 1;

}

if (str2[j] != 0)

{

return -1;

}

return 0;

}

#include<stdio.h>

#define N 20

void main()

{

int i = 0, j = 0;

char str[N], temp;

puts("Please Enter string");

gets(str);

while (str[i] != NULL)

{

j++;

i++;

}

for (i = 0, j = j - 1; i<j; i++, j--)

{

temp = str[i];

str[i] = str[j];

str[j] = temp;

}

printf("%s \n", str);

}

המשתמש מכניס משפט ואנו עוברים ומשנים כל אות ראשונה בכל מילה לאות גדולה (אם היא כבר גדולה- לא נוגעים בה).

תשובה:

#include <stdio.h>

void main()

{

int i,len;

char w1[256];

printf("Please enter a sentence:\n");

gets(w1);

for (i=0; w1[i] ; i++)

if (!i || w1[i-1]==' ')

if (w1[i] >='a' && w1[i]<='z')

w1[i] += 'A' - 'a';// זה בעצם מחסר מ 32 w1[i]

printf("%s\n",w1);

}

המשתמש כותב משפט בן 5 מילים, התוכנית מדפיסה על המסך את המילה הארוכה ביותר במשפט. אין להשתמש בפונקציות strlen ,strcmp ,strcpy.

תשובה:

#include <stdio.h>

void main()

{

int i=0,len,maxWordLoc, maxWordLen=0, curWordLen=0;

char w1[256];

printf("Please enter a sentence consisting of 5 words:\n");

gets(w1);

do{

if (w1[i] == ' ' || w1[i] =='\0')// אם אנחנו בסוף המילה האחרונה או בסוף המשפט: {

if (curWordLen>maxWordLen)

{

maxWordLen = curWordLen;

maxWordLoc = i;

}

curWordLen = 0;

}else

curWordLen++;

}while(w1[i++]);

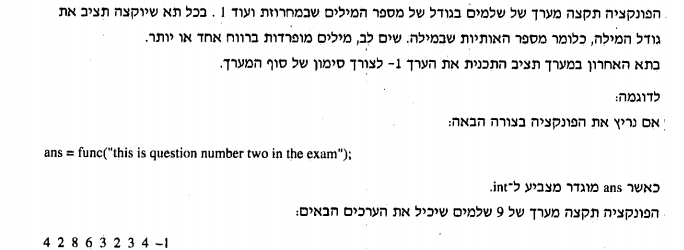
while (maxWordLoc && w1[maxWordLoc-1] !=' ')

maxWordLoc--;

while (w1[maxWordLoc] && w1[maxWordLoc]!=' ')

putchar(w1[maxWordLoc++]);

}

******

#include <stdio.h>

#include <stdlib.h>

int \*func(char \*str)//

{

int i = 0, j = 0, temp=0, count = 0, \*num;

while (str[i] != NULL)//בודק כמה מילים יש במחרוזת לא משנה כמה רווחים יש

{

while (str[i] == ' ')

{

i++;

}

while (str[i] != ' '&& str[i] != NULL)

{

i++;

if (str[i] == ' ' || str[i] == NULL)

{

temp++;

}

}

}

num = (int\*)malloc(sizeof(int)\*temp+1);//הקצאה למערך בגודל כמות המילים +1

i = 0;

while (str[i] != NULL)//בודק את כמות האותיות במילה, ושם את הכמות במערך

{

while (str[i] == ' ')

{

i++;

}

while (str[i] != ' '&& str[i] != NULL)

{

count++;

i++;

}

if (count == 0)//מקרה קצה אם יש רווח לפני ה null

{

count = -1;

}

num[j] = count;

j++;

count = 0;

if (str[i] == NULL)//שם בסוף המערך 1-

{

num[j] = -1;

}

}

return(num);

}

void f(char\*str, int num)//פונקציה שהופכת את המילים במחרוזת

{

if (num == 0)return;

printf("%c", str[num - 1]);

f(str, num - 1);

return;

}

int main()

{

char c, S[100];

int i = 0;

int cnt = 0;

int I\_S = 0;

char string[] = " the number six !#$n ";

while (string[cnt] != '\0')//כל עוד לא הגענו לסוף המחרוזת תכנס

{

c = string[cnt];//שומר את התו הנוכחי

if ((c >= 'a'&&c <= 'z') || (c >= 'A'&& c <= "Z"))//בודקת אם זה אות קטנה או גדולה

{

S[i] = c;

i++;

I\_S = 1;

}

else//אם זה לא אות קטנה או גדולה הוא יהפוך את האותיות

{

if (I\_S)

f(S, i); //נכנס לפנוקציה והופך את האותיות

printf("%c", c);

i = 0;

I\_S = 0;

}

cnt++;

}

if (I\_S)//

f(S, i);

putchar('\n');

}

void func(char \*str)

void func(char \*str) //פונקציה שהופכת אות קטנה לגדולה ואות גדולה לקטנה אם יש משהו שהוא לא אותיות היא מוחקת

{

int i;

for (i = 0; str[i] != '\0'; i++)//בודקת אם זה הגיע לסוף המחרוזת

{

if ((str[i] >= 'a') && (str[i] <= 'z'))//בודק אם זה אות קטנה

{

str[i] += 'A' - 'a';

}

else if ((str[i] >= 'A') && (str[i] <= 'Z'))//בודק אם זה אות גדולה

{

str[i] += 'a' - 'A';

}

else if (str[i] == ' ')//בודק אם זה רווח

{

str[i] = ' ';

}

else if (!(((str[i] >= 'A') && (str[i] <= 'Z')) || ((str[i] >= 'a') && (str[i] <= 'z')))) //בודק אם זה לא אות גודלה או קטנה או רווח הוא מוחק אותה

{

str[i] = '\0';

strcat(str, &str[i + 1]);

i--;

}

}

}

int func(char \*palindr2,int len) // פונקציה בודקת אם המחרוזת פולנדרום בלי להתחשב בסימנים או אותיות גדולות

{

int i, j;

for (i = 0, j = len - 1; i < len, j >= 0; i++, j--)//

{

while (!isalpha(palindr2[i])) i++;//

while (!isalpha(palindr2[j])) j--;//

if ((tolower(palindr2[i]) != tolower(palindr2[j]))) //אם האות גדולה הוא הופך אותה לקטנה

{

printf("%s - It's not a palindrome\n", palindr2);

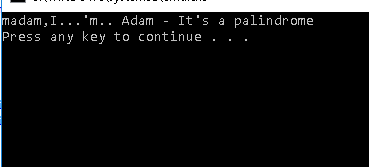
return 0;

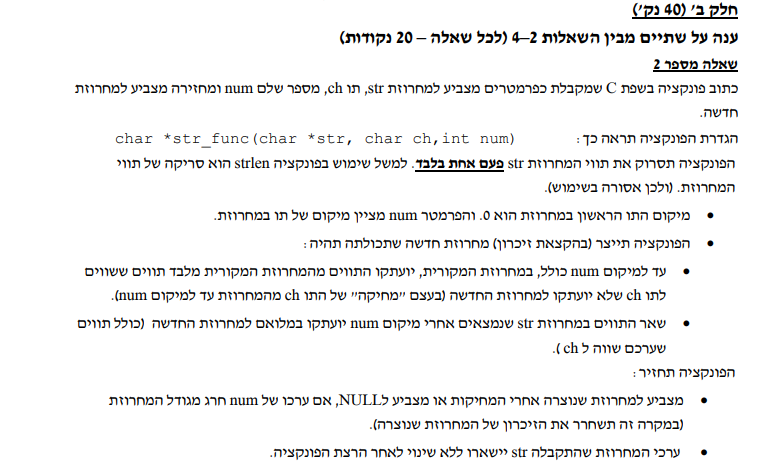
}

}

printf("%s - It's a palindrome\n", palindr2);

}





#include <stdio.h>

#include <string.h>

#include <stdlib.h>

char \*delete\_char\_f(char \*str, char c, int num);

int main()

{

char str[] = "this is question number 2", \*new\_str;

new\_str = delete\_char\_f(str, 'i', 13);

puts(new\_str);

return (0);

}

char \*delete\_char\_f(char \*str, char c, int num)

{

char \*new\_str, ch = c;

int i, k = 0;

new\_str = (char \*)calloc(num, sizeof(char));

for (i = 0; str[i] != NULL; i++)

{

if (i <= num)

if (str[i] != ch)

new\_str[k++] = str[i];

if (i > num)

{

new\_str = (char \*)realloc(new\_str, k + 1);

new\_str[k++] = str[i];

}

}

new\_str[k] = NULL;

return (new\_str);

}

int find\_word\_in\_string(char \*string, char \*word)//פונקציה שבודקת אם המילה בתוך מחרוזת או מוצאת את המיקום של המילה בסטרינג

{

int i, j = 0, count = 0, k = 0;

for (i = 0; i < strlen(string); i++)

{

k = i;

while (string[k] == word[j] && word[j]!=NULL )

{

count++;

k++;

j++;

}

if (count == strlen(word))

{

return 1; //אם נרצה את המיקום של תחילת המילה בסטרינג נחזיר את i

}

else

{

j = 0;

}

count = 0;

}

return 0;

}

void Caesar\_Cipher\_func(char \*arr)//פענוח צופן קיסר

{

int i = 0, j = 0, flag = 0;

char \*NewArr;

NewArr = (char\*)malloc(sizeof(char)\*strlen(arr));

char test[] = "bY", test2[] = "this", test3[] = "the";

for (i = 0; i < 25; i++)

{

for (j = 0; arr[j] != NULL; j++)

{

if (isalnum(arr[j]) == 4)

{

NewArr[j] = arr[j];

continue;

}

if (arr[j] == ' ')

{

NewArr[j] = arr[j];

continue;

}

if (isalnum(arr[j]) == 0)

{

continue;

}

if (isalnum(arr[j]) == 1)

{

if ((arr[j] + i) <= 90 && ((arr[j] + i) >= 65))

{

NewArr[j] = arr[j] + i;

}

else {

flag = arr[j] + i - 26;

NewArr[j] = flag;

flag = 0;

}

}

if (isalnum(arr[j]) == 2)

{

if ((arr[j] + i) >= 97 && ((arr[j] + i) <= 122))

{

NewArr[j] = arr[j] + i;

}

else {

flag = arr[j] + i - 26;

NewArr[j] = flag;

flag = 0;

}

}

}

NewArr[j] = NULL;

if ((strstr(NewArr, test) != 0) || (strstr(NewArr, test2) != 0) || (strstr(NewArr, test3) != 0))

{

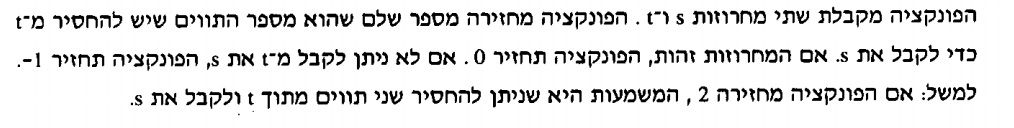
puts("");

puts(NewArr);

}

}

}



#include <stdio.h>

int diff\_str(char \*t, char \*s)

{

if (strlen(t) > strlen(s))//במידה והמחרוזת הימנית ארוכה מהשמאלית

{

return -1;

}

int i = 0, j = 0,count=0;

for ( i = 0; t[i]!=NULL; i++)

{

for (j = 0; s[j] != NULL; j++)

{

if (t[i] == s[j])

{

count++;

break;

}

}

}

if (count != strlen(t))//במידה ולא מצאנו את כל המחרוזת השמאלית

{

return -1;

}

if (count == strlen(t) && count < strlen(s))//במידה ומצאנו את המחרוזת השמאלית, אך הימנית ארוכה יותר – החזרת כמות האותיות שיש להחסיר מהימנית

{

return(strlen(s) - strlen(t));

}

if (count == strlen(s)&& strlen(t)==count)//במידה והמחרוזות זהות

{

return 0;

}

}