Міністерство освіти та науки України  
Національний технічний університет України  
«Київський політехнічний інститут»  
Факультет прикладної математики  
Кафедра системного програмування і спеціалізованих  
комп’ютерних систем

**Лабораторна робота №1**

**Частина 1**з дисципліни

**«Об’єктно орієнтоване програмування»**Тема: **«**Указатели, строки, массивы**»**

Виконав: Сахнік Іван Сергійович

Студент групи КВ-42

Перевірив(ла)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Київ-2015**

2.2. Задание на работу (часть I).

**1)** Написать функцию int substr(const char \*string1, const char \*string2) , которая возвращает индекс элемента в строке string1, с которого начинается подстрока, равная string2.

**2)** Написать функцию int subseq(const char \*string1, const char \*string2), которая возвращает наибольшую длину общей подпоследовательности символов строк string1 и string2. Например, subseq («доцент», «студент») = = 3.]

**3)** Написать функцию char ispal(const char \*string), которая возвращает 1, если string является *палиндромом* и 0 – в противном случае. Палиндромом называется последовательность символов, которая выглядит одинаково будучи прочтенной как справа налево, так и наоборот. Например, “123X321” или “abccba”.

**4)** Написать функцию char\* makepal(const char \*), которая получает в качестве параметра указатель на строку символов, преобразует ее в *палиндром* добавляя к ней *наименьшее* число символов в конец строки и возвращает указатель на палиндром.

**5)** Написать функциию double\* txt2double(const char \* , int \*size). Первый параметр – указатель на строку, в которой размещены числа, разделенные символом “;” , например, “1123.45;13;3.14”. Функция txt2double() размещает эти числа в динамическом массиве и возвращает указатель на него. Параметр size принимает значение, равное длине массива в случае удачного завершения функции и 0 – в противном случае (т.е. если хотябы одно из чисел в строке невозможно корректно преобразовать – “1123.45;Sb3;3.14”).

***Текст програми:***

#ifndef MYSTRING\_H

#define MYSTRING\_H

int substr(const char \*string1, const char \*string2);

int subseq(const char \*string1, const char \*string2);

int inspal(const char \*string);

char\* makepal(const char \*string);

double\* txt2double(const char \*txt, int \*size);

#endif //MYSTRING\_H

#include"mystring.h"

#include<string.h>

#include<stdio.h>

#include<malloc.h>

#include<stdlib.h>

int substr(const char \*string1, const char \*string2)

{

int str2 = strlen(string2);

int str1 = strlen(string1);

int i,j;

int result=-1;

if (str2>str1)

printf("string too large\n");

for (i=0; i<str1 - str2 + 1; i++)

for(j=0; j<str2; j++)

{

if (string2[j]!=string1[i+j])

break;

else

if (j==str2-1)

result=i;

}

return result+1;

}

int inspal(const char \*string)

{

int i;

int len=strlen(string);

for(i=0; i<=len/2+1; i++)

if (string[i]!=string[len-i-1])

return 0;

return 1;

}

int subseq(const char \*string1, const char \*string2)

{

int i,j,n,k,max,tmp;

int s1 = strlen(string1);

int s2 = strlen(string2);

n=0; max=0; tmp=0;

for (i=0; i<s1; i++)

for(j=0; j<s2; j++)

{

if (s2-j<s1-i) n=s2-j;

else n=s1-i;

tmp=-1;

for(k=0; k<n; k++)

if (string1[i+k]!=string2[j+k])

{tmp=k; break;}

if (tmp==-1) tmp=n;

if (tmp>max) max=tmp;

}

return max;

}

char\* makepal(const char\* string)

{

int length=strlen(string);

int L=0, R=length-1, count=0, tmpCount;

char\* pal;

int i,j;

while (L<R)

{

while (string[L]!=string[R])

{

L++;

count++;

}

tmpCount=0;

while (string[L]==string[R])

{

L++; R--; tmpCount++;

}

if (L<R)

{

R=length-1;

count +=tmpCount;

}

}

pal=(char\*)malloc((length+count+1)\*sizeof(char));

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

pal[i]=string[i];

for(j=0; j<count; j++,i++)

pal[i]=string[count-j-1];

pal[i]='\0';

return pal;

}

double\* txt2double(const char \*string, int \*size)

{

int len1, len2, n = 0, j, i, num;

char \*check = "0123456789.;" ;

double \*arr;

char s[100];

len1 = strlen(string);

len2 = strlen(check);

\*size = 0;

for (i = 0; i < len1; i++)

{

if (string[i] == ';'|| string[i+1]=='\0')

{

if (string[i + 1] != ';')

n++;

else

continue;

}

for (j = 0; j < len2; j++)

if (string[i] == check[j])

break;

if (j == len2)

return NULL;

}

arr=(double\*)malloc(n\*sizeof(double));

for (i = 0, j = 0, num=0; i <= len1; i++)

if (string[i] != ';' && string[i]!='\0')

{

s[j] = string[i];

j++;

}

else

{

if (j==0) continue;

s[j]='\0';

j = 0;

arr[num] = atof(s);

num++;

(\*size)++;

while (s[j] != '\0')

{

s[j] = '\0';

j++;

}

j = 0;

}

return arr;

}

#include"mystring.h"

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main()

{

int i,n;

double \*p;

char string[100], string1[100], string2[100], \*string3="3.12;;4.5;";

strcpy(string,"ent");

strcpy(string1,"student");

strcpy(string2,"docent");

printf("substring on the position: %s %s\n",string1,string);

printf("%d\n\n",substr(string1,string));

printf("the longest substring: %s %s\n",string1,string2);

printf("%d\n\n",subseq(string1,string2));

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

if (inspal(string))

printf("string is palindrom\n\n");

else

printf("string not palindrom\n\n");

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

printf("%s\n\n",makepal(string));

p = txt2double(string3, &n);

if (p!=NULL)

{

printf("double count %d\n",n);

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

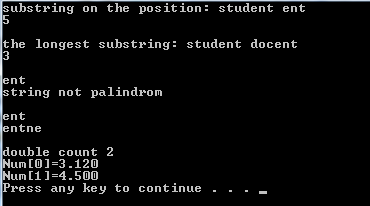
printf("Num[%d]=%.3f\n", i, p[i]);

}

else

printf("String error\n");

system("pause");

******

return 0;

}

***Тестування програми:***