**МИНОБРНАУКИ РОССИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕНЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.Шухова»**

**(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа № 1

По дисциплине: Алгоритмы и СД

Тема: «Производные структуры данных.

Структура данных типа «строка» (Pascal/C)

Выполнил: ст.группы ПВ - 21

Браткова И.О.

Белгород 2017

**Цель работы:** изучение встроенной структуры данных типа «строка», разработка и использование производных структур данных строкового типа.

**Задание**

1. Для СД типа строка определить:

1.1. Абстрактный уровень представления СД:

1.1.1 Характер организованности и изменчивости.

1.1.2. Набор допустимых операций.

1.2. Физический уровень представления СД:

1.2.1. Схему хранения.

1.2.2. Объем памяти, занимаемый экземпляром СД.

1.2.3. Формат внутреннего представления СД и способ его интерпретации.

1.2.4. Характеристику допустимых значений.

1.2.5. Тип доступа к элементам.

1.3. Логический уровень представления СД.

Способ описания СД и экземпляра СД на языке программирования.

2. Реализовать СД строкового типа в соответствии с вариантом индивидуального задания (см. табл.8) в виде модуля. Определить и обработать исключительные ситуации.

3. Разработать программу для решения задачи в соответствии с вариантом индивидуального задания (см. табл.8) с использованием модуля, полученного в результате выполнения пункта 2.

**Задание 1**

Для СД типа строка определить:

1. Абстрактный уровень представления СД:
   1. Характер организованности и изменчивости: **последовательная, динамическая**
   2. Набор допустимых операций: **сравнение, присваивание, конкатенация**
2. Физический уровень представления СД:
   1. Схема хранения: **последовательная**
   2. Объём памяти, занимаемой экземпляром СД: **N+1 байт, где N – максималное количество символов в строке(последний символ ‘\0’ – признак конца строки).**
   3. Формат внутреннего представления СД и способ его интерпретации: **массив символов, с завершающим \0-символом.**
   4. Характеристика допустимых значений: **N символов таблицы ASCII.**
   5. Тип доступа к элементам: **прямой**
3. Логический уровень представления СД:
   1. Способ описания СД и экземпляра СД на языке программирования: **typedef char string[N];**

**Задание 2**

**Form4.h**

#ifndef FORM4\_H\_INCLUDED

#define FORM4\_H\_INCLUDED

//#define OK 0

//#define OUT 1

const int out = 1;

const int ok = 0;

typedef struct

{

char st[1024];

unsigned n; // Длина строки

} string1;

void InputStr(string1\* s); // Ввод строки

void OutputStr(string1 s);//Вывод строки

void WriteToStr (string1 stt, char \*s);//Запись строки s в строку stt

void WriteFromstr(char \*s1, string1 s);//Запись в строку s из строки s1

int CComp(string1 s1, string1 s2); //Сравнение строк

void Delete(string1\* s, unsigned ind, unsigned cou); //Удаление из строки cou символов с позиции ind

void Concat(string1 s1, string1 s2, string1\* srez); //Конкатенация строк

void Copy(string1 s, unsigned Index, unsigned Count, string1\* Subs); );//запись count символов в строку Subs из строки s, начиная с позиции index

unsigned Pos(string1 SubS, string1 s); );// позицию, с которой в строке s располагается подстрока Subs

int StrError; //переменая ошибок

#endif // FORM4\_H\_INCLUDED

**Fom4.c**

#include <stdio.h> #include <stdlib.h> #include <string.h>#include "form4.h"

void InputStr(string1\* s)

{

char ch=0;

int i=0, n=0;

while (ch!='\n')

{

ch = getchar();

(\*s).st[i]=ch;

i++;

}

(\*s).n = i;

(\*s).st[i--] = '\0';

}

void OutputStr(string1 s)

{

int i=0, len = s.n;

for (i=0; (i<len) && (s.st[i]!='\0'); i++) putchar(s.st[i]);

}

void WriteToStr (string1 stt, char \*s) {

int i = 0;

while (\*s++!= '\0')

{

stt.st[i]=s[i];

i++;

}

}

void WriteFromstr(char \*s1, string1 s) {

int i = 0;

int len = s.n;

if (len > 255)

{

**StrError = OK;**

while (s.st[i] != '\0')

{

s1[i] = s.st[i];

i++;

}

s1[i] = '\0'; return;

}

else

{

**StrError = OUT;**

**printf("Запись не может быть произведена!");**

return 0;

}

}

int CComp(string1 s1, string1 s2)

{

int i;

for (i = 0; s1.st[i] == s2.st[i]; i++) if (s1.st[i] == '\0') return 0;

return s1.st[i] - s2.st[i];

}

void Delete(string1\* s, unsigned ind, unsigned cou)

{

int i=0, k=s->n;

for (i = ind; i < k; i++)

{

s->st[i] = s->st[cou+1]; cou++;

}

}

void Insert(string1 SubS, string1\* s, unsigned Index)

{

int i, j = 0, k=s->n;

int len = SubS.n;

char c;

for (i=k+len; i > Index+len; i--)

{

s->st[i] = s->st[i-len];

for (i=Index+2; i<=Index+len; i++)

s->st[i] = SubS.st[i-Index];

}

}

void Concat(string1 s1, string1 s2, string1\* srez)

{

unsigned i=0;

(\*srez).n=s1.n+s2.n;

if ((\*srez).n < s1.n+s2.n)

{

**StrError = OUT;**

**printf("Результирующуая длина меньше!");**

return 0;

}

else

{

**StrError = OK;**

while (i<s1.n)

{

(\*srez).st[i] = s1.st[i];

i++;

}

while (i<s2.n + s1.n)

{

(\*srez).st[i] = s2.st[i-s1.n];

i++;

}

(\*srez).st[i]='\0';

return;

}

}

void Copy(string1 s, unsigned Index, unsigned Count, string1\* Subs)

{

int i;

if (Subs->n < Count)

{

**StrError = OK;**

for (i = Index + 2; i <= Index+Count+2; i++)

{

Subs->st[i-Index] = s.st[i];

Subs->st[i-Index+1] = '\0';

}

return;

}

else

{

**StrError = OUT;**

**printf("Строка не может быть скопирована!");**

return 0;

}

}

unsigned Pos(string1 SubS, string1 s) //возврвщает позицию, с которой в строке s располагается подстрока SubS

{

int i=0, j=0;

while (SubS.st[i] != '\0')

{ if (SubS.st[i] == s.st[i]) return i;

j++;

}

return 0;

}

**Задание 3**

Поиск последнего вхождения подстроки s1 в строку s2.

unsigned LastPos(string1 s1, string1 s2)

{

int l1=s1.n;

int l2=s2.n;

int i;

if (l1-l2 < 0)

{

**SrtError = OUT;**

**printf("Подстрока не может быть длинее строки!");**

return 0;

}

else

{

**StrError = OK;**

for(i=l1-l2;i>=0;i--)

{

int j=0;

for(;j<l2;j++)

if(s1.st[i+j]!=s2.st[j])

break;

if(j==l2)

return i+1;

}

return 0;

}

}

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "form4.h"

int main()

{

string1 s1,s2,s3;

InputStr(&s1);

InputStr(&s2);

unsigned ch = LastPos(s1, s2);

printf("%d", ch);

return 0;

}

 