**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №2

дисциплина: Алгоритмы и Структуры данных

тема: «Производные структуры данных.»

Выполнил: ст. группы ПВ-21

Донцов Александр Алексеевич

Проверил: Синюк Василий Григорьевич

Белгород 2018

**Структура данных типа «строка» (Pascal/C)**

**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.

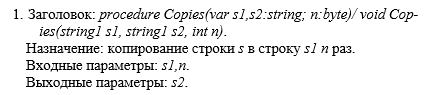
1.2.5. Тип доступа к элементам.

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.



**Формат 1**

**Реализация на языке C:**

#if !defined(\_\_FORM1\_H)

#define \_\_FORM1\_H

const ...; // Определение исключительных ситуаций

typedef char string1[256];

// Признак конца строки - символ '\0'

void WriteToStr(string1 st, char \*s);

void WriteFromStr(char \*s, string1 st);

void InputStr(string1 s);

void OutputStr(string1 s);

int Comp(string1 s1, string1 s2);

void Delete(string1 s, unsigned Index, unsigned Count);

void Insert(string1 Subs, string1 s, unsigned Index);

void Concat(string1 s1, string1 s2, string1 srez);

void Copy(string1 s, unsigned Index, unsigned Count, string1 Subs);

unsigned Length(string1 s);

unsigned Pos(string1 SubS, string1 s);

int StrError; // Переменная ошибок

//...

#endif

**Выполнение**

1. Для СД типа строка определить:

1.1. Абстрактный уровень представления СД:

1.1.1 Характер организованности и изменчивости.

*представляет собой линейную структуру — последовательность*

1.1.2. Набор допустимых операций.

- операция присваивания

- операция сравнения

- операция конкатенация

1.2. Физический уровень представления СД:

1.2.1. Схему хранения.

На физическом уровне строка реализована последовательной схемой хранения.

1.2.2. Объем памяти, занимаемый экземпляром СД.

зависит от максимального количества элементов (символов) в строке и определяется формулой:

Vстр = K + 1, где K — максимальное количество символов в строке.

1.2.3. Формат внутреннего представления СД и способ его интерпретации.

*Может храниться в статической или динамической памяти Массив типа char размером K+1(+1 для признака конца строки)*

1.2.4. Характеристику допустимых значений.

*СAR(string) = 1 + 256 + 2562 + ... + 256K , где K — максимальное количество элементов в строке.*

1.2.5. Тип доступа к элементам.

*прямой, осуществляется по индексу*

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.

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdlib.h"

#include "stdio.h"

#include "pch.h"

#include <iostream>

// lab\_2.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

//#if !defined(\_\_FORM1\_H)

// #define \_\_FORM1\_H

const int MAXSTRLN = 256;

const int OK = 0; // Определение исключительных ситуаций

const int NO\_ROWS = 1; // Определение исключительных ситуаций

const int EXCEEDED\_STR = 2; // Определение исключительных ситуаций

const int NO\_CHAR = 3; // Определение исключительных ситуаций

typedef char string1[256];

// Признак конца строки - символ '\0'

void WriteToStr(string1 st, char \*s);

void WriteFromStr(char \*s, string1 st);

void InputStr(string1 s);

void OutputStr(string1 s);

int Comp(string1 s1, string1 s2);

void Delete(string1 s, unsigned Index, unsigned Count);

void Insert(string1 Subs, string1 s, unsigned Index);

void Concat(string1 s1, string1 s2, string1 srez);

void Copy(string1 s, unsigned Index, unsigned Count, string1 Subs);

unsigned Length(string1 s);

unsigned Pos(string1 SubS, string1 s);

void Copies(string1 s1, string1 s2, int n);//копирование строки s в строку s1 n раз.

int StrError; // Переменная ошибок

//...

//#endif

int main()

{

string1 s, s1;

InputStr(s);

Copies(s, s1, 4);

OutputStr(s1);

printf("\nStrError = %d", StrError);

return 0;

}

void Copies(string1 s, string1 s1, int n) {//копирование строки s в строку s1 n раз.

int i;

int len = Length(s);

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

Insert(s, s1, len \* i);

}

s1[len\*i] = '\0';

return;

}

void WriteToStr(string1 st, char \*s) {//Запись данных в строку st из строки s. Строка s заканчивается нулевым символом '\0'.

int i = 0;

if ((st == NULL) || (s == NULL)) {

StrError = NO\_ROWS;

return;

}

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

st[i] = s[i++];

}

st[i] = '\0';

StrError = OK;

}

void WriteFromStr(char \*s, string1 st) {//Запись данных в строку s из строки st. Строка s заканчивается нулевым символом '\0'.

int i = 0;

if ((st == NULL) || (s == NULL)) {

StrError = NO\_ROWS;

return;

}

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

s[i] = st[i++];

}

s[i] = '\0';

StrError = OK;

}

void InputStr(string1 s) {//Ввод строки s с клавиатуры.(проверено)

int i = 0;

char c;

while ((c = getchar()) != '\n') {

if (i > MAXSTRLN) {

StrError = EXCEEDED\_STR;

return;

}

s[i++] = c;

}

s[i] = '\0';

StrError = OK;

}

void OutputStr(string1 s) {//Вывод строки s на экран монитора.(проверено)

int i = 0;

if (s == NULL) {

StrError = NO\_ROWS;

return;

}

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

printf("%c", s[i++]);

}

printf("\n");

StrError = OK;

}

int Comp(string1 s1, string1 s2) {//Сравнивает строки s1 и s2. Возвращает 0 если s1 = s2; 1, если s1 > s2; -1, ес-ли s1 < s2.

int i;

if ((s1 == NULL) || (s2 == NULL)) {

StrError = NO\_ROWS;

return NULL;

}

StrError = OK;

for (i = 0; s1[i] != '\0' || s2[i] != '\0'; i++) {

;

}

if (s1[i] == s2[i])

return 0;

else if (s1[i] == '\0')

return -1;

else

return 1;

return 1;

}

void Delete(string1 s, unsigned Index, unsigned Count) {//Удаляет Count символов из строки s, начиная с позиции Index.

int len = Length(s);

if ((s == NULL)) {

StrError = NO\_ROWS;

return;

}

if ((Count + Index) > len) {

StrError = NO\_CHAR;

return;

}

unsigned score = Count, i = Index;

while (score-- != 0)

s[i] = s[i++ + Count];

StrError = OK;

}

void Insert(string1 Subs, string1 s, unsigned Index){//Вставляет подстроку SubS в строку s, начиная с позиции Index

unsigned i = Index, j = 0;

if ((Subs == NULL) || (s == NULL)) {

StrError = NO\_ROWS;

return;

}

unsigned len1 = Length(Subs);

unsigned len = Length(s);

if ((len1 + Index) >= MAXSTRLN) {

StrError = EXCEEDED\_STR;

return;

}

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

s[i++] = Subs[j++];

if (len < i)

s[i] = '\0';

StrError = OK;

}

void Concat(string1 s1, string1 s2, string1 srez) { //Выполняет конкатенацию строк s1 и s2. Результат помещает в srez.

unsigned i = 0, j = 0;

if ((s1 == NULL) || (s2 == NULL)) {

StrError = NO\_ROWS;

return;

}

unsigned len1 = Length(s1);

unsigned len2 = Length(s2);

if (len1 + len2 >= MAXSTRLN) {

StrError = EXCEEDED\_STR;

return;

}

while (s1[i] != '\0')

srez[i] = s1[i++];

while (s2[j] != '\0')

srez[i++] = s1[j++];

srez[i] = '\0';

StrError = OK;

}

void Copy(string1 s, unsigned Index, unsigned Count, string1 Subs) {//Записывает Count символов в строку Subs из строки s, начиная с позиции Index.

unsigned score = Count, i = Index, j;

if ((s == NULL) || (Subs == NULL)) {

StrError = NO\_ROWS;

return;

}

if (Count + Index >= MAXSTRLN) {

StrError = EXCEEDED\_STR;

return;

}

for (j = 0; score != 0; j++)

Subs[j] = s[i++];

StrError = OK;

}

unsigned Length(string1 s) {//Возвращает текущую длину строки S.

unsigned count = 0, i = 0;

if (s == NULL) {

StrError = NO\_ROWS;

return NULL;

}

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

count++;

StrError = OK;

return count;

}

unsigned Pos(string1 SubS, string1 s) {//Возвращает позицию, начиная с которой в строке s располагается под-строка SubS.

unsigned rez = 0, i, j;

if ((SubS == NULL) || (s == NULL)) {

StrError = NO\_ROWS;

return NULL;

}

StrError = OK;

while (s[rez] != '\0') {

i = rez;

j = 0;

while (s[i++] == SubS[j])

if (SubS[j++] == '\0')

return rez;

}

return NULL;

}

