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

Дисциплина: Алгоритмы и структуры данных

Лабораторная работа №2

Встроенные структуры данных (Pascal/C)

|  |  |
| --- | --- |
|  | Выполнила: ст. группы ПВ-21  Зановская А.И.  Проверил: Синюк В.Г. |

Белгород

2017

Л а б о р а т о р н а я р а б о т а № 2

Производные структуры данных.

Структура данных «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.

последовательность, динамическая структура.

1.1.2. Набор допустимых операций.

Присваивание, сравнение, конкатенация, инициализация.

1.2. Физический уровень представления СД:

1.2.1. Схему хранения.

Последовательная.

1.2.2. Объем памяти, занимаемый экземпляром СД.

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

1.2.3. Формат внутреннего представления СД и способ его интерпретации.

Mассив символов из K + 1 элементов типа char. Нумеруются элементы от нуля до K, во внутреннем представлении этот массив (строка) заканчивается нулевым символом '\0', по которому программа может найти конец, т.е. элемент с индексом K равен нулю как признак конца строки.

1.2.4. Характеристику допустимых значений.

Каждый элемент представляет собой символ с кодом 0..256

1.2.5. Тип доступа к элементам.

Прямой

1.3. Логический уровень представления СД.

Способ описания СД и экземпляра СД на языке программирования.

1. #define STRLENGTH ...// Значение

...

char s[STRLENGTH];

2. typedef char t\_str10[10];

...

t\_str10 s;

3. typedef char\* p\_str;

...

p\_str ps;

4.typedef char t\_str10[10]:

typedef t\_str10\* p\_str10;

...

p\_str10 ps;

1. Реализовать СД строкового типа в соответствии с вариантом индивидуального задания (см. табл.8) в виде модулей на языках Pascal и С. Определить и обработать исключительные ситуации.

#ifndef \_\_FORM1\_H\_INCLUDED

#define \_\_FORM1\_H\_INCLUDED

typedef char string1[256];

// Признак конца строки - символ '\0'

int StrErr=0;

//Запись данных в строку st из строки s. Строка s заканчивается нулевым символом '\0'.

void WriteToStr(string1 st, char \*s)

{

while (\*st++ = \*s++);

}

//Запись данных в строку s из строки st. Строка s заканчивается нулевым символом '\0'.

void WriteFromStr(char \*s, string1 st)

{

while (\*s++ = \*st++);

}

//Ввод строки s с клавиатуры.

void InputStr(string1 s)

{

gets(s);

}

//Вывод строки s на экран монитора.

void OutputStr(string1 s)

{

puts(s);

}

// Сравнивает строки s1 и s2. Возвращает 0 если s1 = s2; 1, если s1 > s2; -1, если s1 < s2.

int Comp(string1 s1, string1 s2)

{

for (;\*s1 == \*s2; s1++, s2++)

if (\*s1 == '\0')

return 0;

return \*s1 - \*s2;

}

//Удаляет Count символов из строки s, начиная с позиции Index.

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

{

int i;

if (i+Count>Length(s))

{

StrErr=1;

StErr();

}

for (i=Index;s[i];i++)

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

}

//Вставляет подстроку SubS в строку s, начиная с позиции Index.

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

{

unsigned i,j,l1,l2;

l1=Length(s);

l2=Length(Subs);

if (l1<l2+Index-1)

{

StrErr=2;

StErr();

}

else

{

s[l1+l2]='\0';

for (i=l1+l2-1,j=Index+l2-1;i>Index-1;i--,j--)

s[i]=s[j];

for (i=Index,j=0;j<l2;i++,j++)

s[i]=Subs[j];

}

}

//Выполняет конкатенацию строк s1 и s2. Результат помещает в srez.

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

{

if (Length(s1)+Length>255)

{

StrErr=3;

StErr();

}

while (\*srez++ = \*s1++);

\*srez--;

while (\*srez++ = \*s2++);

}

//Записывает Count символов в строку Subs из строки s, начиная с позиции Index.

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

{

int i,l,j;

if (Index+Count<255)

{

StrErr=4;

StErr();

}

for (i=Index,j=0;i<=Count;i++,j++)

Subs[i]=s[j];

}

//Возвращает текущую длину строки S

int Length(string1 s)

{

int i=0;

while (s[i])

i++;

return i;

}

//Возвращает позицию, начиная с которой в строке s располагается под-строка SubS.

unsigned Pos(string1 SubS, string1 s)

{

int i=0,j=0,k,fl=1;

while (s[i])

{

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

i++;

if (!s[i] || i+Length(SubS)<255)

{

StrErr=1;

StErr();

}

k=i++;

for (j++,i++;SubS[j] &&s[i] && fl;i++,j++)

fl=s[i]==SubS[j];

if(fl)

break;

i=k;

j=0;

fl=1;

}

return k;

}

#endif // \_\_FORM1\_H\_INCLUDED

3. Разработать программы на языках Pascal и С для решения задачи в соответствии с вариантом индивидуального задания (см. табл.8) с использованием модулей, полученных в результате выполнения пункта 2.

Заголовок: function StrCSpn(s,s1:string):word/ unsigned StrCSpn(string1 s, string1 s1). Назначение: нахожде­ние длины той части строки s, которая не сoдержит символы из строки s1.

Входные параметры: s,s1.

Выходные параметры: нет.

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include "\_\_FORM1.H"

//нахождение длины той части строки s, которая не сoдержит символы из строки s1.

void main ()

{

setlocale (LC\_ALL, "Rus");

string1 s,s1;

unsigned k;

InputStr(s);

InputStr(s1);

k=StrCSpn(s,s1);

printf ("k = %d\n",k);

StErr();

}

int StrCSpn(string1 s, string1 s1)

{

int i,j,k=0,fl=1;

for (i=0;s[i]!=0;i++)

{

for (j=0;s1[j]!=0 && fl;j++)

{

fl=(s[i]!=s1[j]);

}

if (fl)

k++;

fl=1;

}

return k;

}

void StErr ()

{

switch (StrErr)

{

case 0: printf ("\nОшибок не произошло\n");

exit(StrErr);

case 1: printf ("\nОшибка в функции Delete\n");

exit(StrErr);

case 2: printf ("\nОшибка в функции Insert\n");

exit(StrErr);

case 3: printf ("\nОшибка в функции Concat\n");

exit(StrErr);

case 4: printf ("\nОшибка в функции Copy\n");

exit(StrErr);

case 5: printf ("\nОшибка в функции Pos\n");

exit(StrErr);

}

}