**МИНОБРНАУКИ РОССИИ**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

по теме «Встроенные структуры данных»

Выполнил: ст. группы ВТ-22  
Макаров Даниил Сергеевич

Проверил: Синюк В.Г.

**Белгород 2018**Лабораторная работа № 2

**Производные структуры данных.**

**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.

**Задание варианта 7**

Заголовок: function SudWord(s:string;n:word):string/ string1 \*SudWord(char \*s, unsigned n).

Назначение: выделение из строки s слов, начиная с номера n.

Входные параметры: s,n.

Выходные параметры: нет.

**Формат №7**

#if !defined(\_\_FORM7\_H)

#define \_\_FORM7\_H

const ...; // Определение исключительных ситуаций

typedef struct str

{

char \*s; /\* Указатель на строку. Первые два байта строки s содержат динамическую длину строки \*/

unsigned max; /\* Максимальное количество символов в строке, определяющееся при инициализации \*/

};

typedef str \*string1;

void InitStr(string1 st, unsigned n);

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

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

void InputStr(string1 st);

void OutputStr(string1 st);

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 DoneStr(string1 s);

int StrError; // Переменная ошибок//...

#endif

Исходный код модуля form\_7.c

#include "form\_7.h"

#include <stdlib.h>

#include <stdio.h>

const int STR\_OVER= 100;

const int STR\_EMPTY= 101;

const int STR\_OUTBORDER= 102;

string1 InitStr(unsigned n)

{

string1 st;

st = (string1) malloc(sizeof(string1));

st->s = (char\*) malloc(n+3);

st->max = n;

setLenght(0,st);

st->s[2]=0;

return st;

};

void WriteToStr(char\* oS,string1 st){

unsigned short i=0;

unsigned short j=Length(st)+2;

char flag=1;

while((oS[i]!='\0')&&(flag)){

st->s[i+j]=oS[i];

i++;

};

st->s[i+j]=' ';

if((i+j-2)<st->max){

setLenght(i+j-2,st);

}

else{

StrError=100;

exit(StrError);

}

};

void setLenght(unsigned short size,string1 st){

if(size<st->max){

st->s[0]=size>>8;

st->s[1]=size&255;

}

else{

StrError=100;

exit(StrError);

}

};

void InputStr(string1 st){

unsigned short i= 2;

while(((st->s[i]=getchar())!='\n')&&(i<st->max)){

i++;

}

if(i>st->max){

StrError=100;

printf("ERROR %d",StrError);

exit(StrError);

}

setLenght(i-2,st);

}

void OutputStr(string1 st){

unsigned short i=2;

unsigned short j=Length(st);

if(j==0){

StrError=101;

exit(StrError);

}

else{

while(putchar(st->s[i])!='\0') i++;

}

};

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

unsigned short j=2,i=Index+Count;

unsigned short size=Length(s);

if(size+2<Index){

StrError=102;

exit(StrError);

}

else{

while(s->s[i+j]!=0){

s->s[Index+j]=s->s[i+j];

j++;

}

}

setLenght(size-j+2,s);

}

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

unsigned short s1,s2;

int i,j=0;

s1=Length(Subs);

s2=Length(s);

if (s->max>=s1+s2){

StrError=100;

exit(StrError);

}

else{

for(i=2;i<s1;i++)

{ s->s[Index+s1+i]=s->s[Index+i];

s->s[Index+j]=Subs->s[j];

j++;

}

}

}

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

unsigned short t1,t2;

int i;

t1=Length(s1);

t2=Length(s2);

if (srez->max>=t1+t2){

StrError=100;

exit(StrError);

}

else{

setLenght(t1+t2,srez);

for(i = 2;i<t1+2;i++) srez->s[i]=s1->s[i];

for(int j = 2;j<t2+2;j++){

srez->s[i]=s2->s[j];

i++;

}

srez->s[i]=0;

}

}

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

unsigned short size=Length(s);

int i;

if(size+2<Index){

StrError=102;

exit(StrError);

}

else{

if (Subs->max>=Count){

StrError=100;

exit(StrError);

}

else{

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

s->s[Index+i]=Subs->s[i];

}

}

Subs->s[i]=0;

}

}

int Comp(string1 s1, string1 s2){

int i,r=0;

while((r==0)&&(i!=s1->max)&&(i!=s2->max))

{

r=s1->s[i]-s2->s[i];

i++;

}

return r;

};

void DoneStr(string1 st){

free(st->s);

free(st);

};

unsigned short Length(string1 st){

char a=st->s[0];

char b=st->s[1];

unsigned short c = (((short)a) << 8) | b;

return c;

};

**Исходный код main.c**

#include <stdio.h>

#include <stdlib.h>

#include "form\_7.h"

#include <string.h>

char\* get\_word(char\* string, char\* word){

for (; \*string!='\0' && \*string == ' '; string++);

for (; \*string!='\0' && \*string != ' '; string++){

\*word=\*string;

word++;

}

\*word = '\0';

return string;

}

string1 SubWord(char \*s, unsigned n)

{

string1 t;

int i;

t=InitStr(90);

char word[99];

for(i=1;i<n;i++){

s=get\_word(s,word);

}

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

s=get\_word(s,word);

WriteToStr(word,t);

t->s[Length(t)+3]=' ';

setLenght(Length(t)+1,t);

}

return t;

}

int main()

{

int n;

string1 s1,s2;

s1=InitStr(90);

char str[90];

printf("Input n - ");

scanf("%i",&n);

fflush(stdin);

printf("Input str - ");

gets(str);

s1=SubWord(str,n);

OutputStr(s1);

setLenght((unsigned short)5,s1);

}