Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Филиал

«Минский радиотехнический колледж»

Учебная дисциплина «Структуры и алгоритмы обработки данных»

Подготовил:

Тамкович Д. И., гр. 62492

Минск

2018**Лабораторная работа № 5**

**Тема работы: «Реализация стека и основных алгоритмов его обработки»**

**1. Цель работы**

Изучение принципов работы со структурой данных стек, формирование умений применения структуры для вычисления арифметических выражений.

**2. Задание**

1. Используя стек, напечатать содержимое текстового файла, выписывая литеры каждой его строки в обратном порядке.

**Type**

uk=^dspis;

dspis=**record**

inf :integer;

next: uk;

**end**;

**Var**

f:textfile;

stek,x:uk;

**Begin**

assignfile(f,'input.txt'); reset(f);

stek:=nil;

**while not** eof(f) **do**

**begin**

**new**(x);

read(f,x^.inf);

x^.next:=stek;

stek:=x;

**end**;

closefile(f);

**while** stek<>nil **do**

**begin**

write(stek^.inf,' ');

stek:=stek^.next;

**end**;

**End**.

2. Выполнить задание в соответствии с вариантом:

**Вариант 1.** Разместить в стеке 10 символов и распечатать их в обратном порядке.

**Type**

uk=^dspis;

dspis=**record**

inf :integer;

next: uk;

**end**;

**Var**

f:textfile;

stek,x:uk;

**Begin**

assignfile(f,'input.txt'); reset(f);

stek:=nil;

**while not** eof(f) **do**

**begin**

**new**(x);

read(f,x^.inf);

x^.next:=stek;

stek:=x;

**end**;

closefile(f);

**while** stek<>nil **do**

**begin**

write(stek^.inf,' ');

stek:=stek^.next;

**end**;

**End**.

3. С использованием стека вычислить арифметическое выражение, предварительно преобразовав его в постфиксную форму записи.

**Type**

pt\_1=^el\_1;

el\_1=**Record**

data:Char;

next:pt\_1;

**End**;

pt\_2=^el\_2;

el\_2=**Record**

data:Real;

next:pt\_2;

**End**;

**Var**

s:String;

f:Boolean;

rez\_1:String;

rez\_2:Real;

**Procedure** WriteStack\_1(**Var** u:pt\_1; dig:Char);

**Var** x: pt\_1;

**Begin**

**New**(x);

x^.data:=dig;

x^.next:=u;

u:=x;

**End**;

**Function** Free\_1(u:pt\_1):Boolean;

**Begin**

**If** u=Nil **Then** Free\_1:=False

**Else**

Free\_1:= True;

**End**;

**Procedure** ReadStack\_1(**Var** u:pt\_1; **Var** dig:Char);

**Var** x: pt\_1;

**Begin**

dig:=u^.Data;

x:=u;

u:=u^.Next;

Dispose(x);

**End**;

**Function** Priority(c: Char): Byte;

**Var** R:Byte;

**Begin**

R := 0;

**Case** c **of**

'+', '-': R:= 1;

'\*', '/': R:= 2;

**end**;

Priority:=R;

**End**;

**Procedure** Convert\_expression(a:String;**Var** z:String);

**Var**

head:pt\_1;

i:Integer;

w:Char;

**Begin**

head:=Nil;

z:='';

i:=1;

**While** i<=Length(a) **Do**

**Begin**

**Case** a[i] **of**

'0'..'9': z:=z+a[i];

'(': WriteStack\_1(head,a[i]);

')': **Begin**

ReadStack\_1(head,w);

**While** w<>'(' **Do**

**Begin** z:=z+w;

ReadStack\_1(head,w);

**End**;

**End**;

'+','-','\*','/':

**Begin**

**If not** Free\_1(head) **Then** WriteStack\_1(head,a[i])

**else**

**begin**

w:=head^.data;

**While** Free\_1(head)**and**(Priority(head^.data)>= Priority(a[i])) **do**

**Begin**

ReadStack\_1(head,w);

z:=z+w;

**End**;

WriteStack\_1(head,a[i]) ;

**end**;

**end**;

**End**;

Inc(i);

**End**;

**While** Free\_1(head) **Do**

**Begin**

ReadStack\_1(head,w);

z:=z+w;

**End**;

**End**;

**Procedure** WriteStack\_2(**Var** u:pt\_2; dig:Real);

**Var** x: pt\_2;

**Begin**

**New**(x); x^.data:=dig;

x^.next:=u;

u:=x;

**End**;

**Procedure** ReadStack\_2(**Var** u:pt\_2; **Var** dig:Real);

**Var** x: pt\_2;

**Begin**

dig:=u^.Data;

x:=u;

u:=u^.Next;

Dispose(x);

**End**;

**Function** Free\_2(u:pt\_2):Boolean;

**Begin**

**If** u=Nil **Then** Free\_2:=False

**Else**

Free\_2:= True;

**End**;

**procedure** Operation(x:char; fm,sm:real; **Var** turn:real);

**begin**

**case** x **of**

'+': turn:=fm+sm;

'-': turn:=fm-sm;

'\*': turn:=fm\*sm;

'/': turn:=fm/sm;

**end**;

**end**;

**Procedure** Convert\_calculate(a:String;**Var** pp:Boolean;**Var** z:Real);

**Var** head: pt\_2;

i,k: Integer;

r,w: Real;

**Begin**

head:=Nil;

pp:=True;

i:=1;

**While** (i<=Length(a)) **And** pp **Do**

**Begin**

**If** a[i]<>' ' **Then Begin**

**If Not**(a[i] **In** ['+','-','\*','/'])

**Then Begin**

Val(a[i],r,k);

**If** k=0 **Then** WriteStack\_2(head,r) **Else** pp:=False;

**End**

**Else Begin**

**If** Free\_2(head) **Then** ReadStack\_2(head,r) **Else** pp:=False;

**If** Free\_2(head) **Then** ReadStack\_2(head,w) **Else** pp:=False;

**If** pp **Then Begin**

Operation(a[i],r,w,r);

WriteStack\_2(head,r);

**End**;

**End**;

**End**;

Inc(i);

**End**;

**If** Free\_2(head) **Then** ReadStack\_2(head,z) **Else** pp:=False;

**End**;

**Var**

f1:textfile;

zzzz:string;

**Begin**

assignfile(f1,'input.txt'); reset(f1);

read(f1,s);

zzzz:='';

Convert\_expression(s,rez\_1);

writeln(rez\_1);

closefile(f1);

Convert\_calculate(rez\_1,f,rez\_2);

writeln(rez\_2);

**end**.

**Контрольные вопросы и задания**

1. Что такое стековый принцип сохранения элементов?

В стеке реализуется дисциплина обслуживания LIFO:

**LAST** – последний; **INPUT** – вошел; **FIRST** – первый; **OUTPUT** – вышел.

2. Какие основные операции реализуются для стеков?

*- инициализация стека;*

*- включение нового элемента*

*- исключение элемента из стека*

3. Какие шаги выполняются при добавлении элемента в стек-массив?

*Процедура добавления элемента в стек должна содержать два параметра: первый задает вершину стека, в который нужно занести элемент, второй – заносимое значение элемента стека.*

4. Какие шаги выполняются при удалении элемента из стека-массива?

*В результате выполнения этой операции некоторой переменной* ***i*** *должно быть присвоено значение первого элемента стека, и значение указателя на начало списка должно быть перенесено на следующий элемент стека.*

5. Какую структуру должен иметь элемент стека при динамической реализации?

*Структуру линейного однонаправленного списка*

6. Как между собой связываются соседние элементы стека?

*В стеке соседние элементы связаны указателями.*

7. Какие типы данных необходимы для динамической реализации стека?

*Обычно динамические структуры описываются через запись, строковые типы данных или числовые.*

8. Какие переменные необходимы для реализации операций с динамическим стеком?

uk=^dspis;

dspis=**record**

inf :integer; - *информационный блок*

next: uk; - *указатель*

**end**;

9. Приведите достоинства и недостатки динамической и статической реализации стека.

*Достоинством статической реализации является простота реализации, а недостатком неравномерное использование памяти. В статической реализации минусом является сложность написания программы, но зато происходит равномерное потребление ресурсов.*