***Министерство образования Республики Беларусь***

***Учреждение Образования***

***«Брестский Государственный Технический Университет»***

***Кафедра ИИТ***

**Лабораторная работа №14**

**По дисциплине ОАиП за II семестр**

**Тема: «Динамические структуры. Стек. Список»**

**Выполнил:**

Студент 1-го курса

Группы ИИ-15(1)

Волк И. А.

**Проверила:**

Хацкевич М. В.

Брест 2018

Цель: научиться работать со стеком и списками.

**Задание 1.**

10. В текстовом файле без ошибок записано логическое выражение следующего вида:

<лог.выр>::=true | false | <лог.выр> and <лог.выр> | <лог.выр> or <лог.выр>

Используя стек, вычислить значение этого выражения с учетом общепринятого приоритета операций.

Код программы:

#include "stdafx.h"

#include <Windows.h>

#include <fstream>

#include <iostream>

#include <cmath>

#include <string>

using namespace std;

struct ex

{

bool content;

ex \* next = NULL;

};

bool boolExFromFile(const char \*);

ex \* exFromStr(char\*);

void error(const char\*, int);

void push(ex \*, bool);

bool pop(ex \*);

int main()

{

cout << boolExFromFile("text.txt") << endl;

system("pause");

return 0;

}

bool boolExFromFile(const char \* nameOfFile)

{

ifstream fin(nameOfFile);

if (!fin)

{

error("Cannot open file or it is empty!", 1000);

return 0;

}

char str[1024];

string s;

getline(fin, s);

strcpy\_s(str, sizeof(str), s.c\_str());

bool res = false;

ex \* ex = exFromStr(str);

while (ex->next!=NULL)

{

res |= pop(ex);

}

res |= pop(ex);

return res;

}

ex \* exFromStr(char\* str)

{

ex \* exNow = new ex; int numOfEx = 0;

bool conNow = true;

int i = 0; int endOfLastCon = -1;

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

{

if (str[i] == ' ') { i++; continue; }

if (!(str[i] - 48) || !(str[i] - 49))

conNow = (int)str[i] - 48;

if (str[i] == 'a')

{

i += 4;

conNow &= (int)str[i] - 48;

endOfLastCon = i;

}

if (str[i] == 'o')

{

if (!numOfEx) exNow->content = conNow;

push(exNow, conNow);

conNow = true;

numOfEx++;

}

if (str[i + 1] == '\0')

{

if (!numOfEx) exNow->content = conNow;

else exNow = push(exNow, conNow);

}

i++;

}

return exNow;

}

void push(ex \* stack, bool content)

{

ex \* temp = new ex;

temp->content = stack->content;

temp->next = stack->next;

stack->next = temp;

stack->content = content;

}

bool pop(ex \* stack)

{

if (stack->next == NULL)

return stack->content;

bool cont = stack->content;

stack->content = stack->next->content;

stack->next = stack->next->next;

return cont;

}

void error(const char\* content, int timer)

{

system("cls");

cout << content << endl;

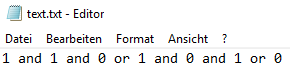
Sleep(timer);

system("cls");

}

Результат:

Содержимое файла:



Выполнение:



**Задание 2.**

9. Написать программу, содержащую процедуры формирования и просмотра списка со информационной частью являющейся произвольным массивом. Включите в нее функцию, которая удаляет из информационной части отрицательные числа.

Код программы:

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

#include <cmath>

using namespace std;

struct list

{

int size;

int \* arr;

list \* next = NULL;

};

list \* init();

void addElement(list \*);

void outList(list \*);

void deleteNegNumsFromList(list \*);

int \* deleteFromArrNeg(int\*, int &);

int main()

{

list \* list = init();

for(int i = 0; i < 4; i++)

addElement(list);

outList(list);

cout << endl;

deleteNegNumsFromList(list);

outList(list);

cout << endl;

system("pause");

return 0;

}

list \* init()

{

list \* newList = new list;

int sizeOfContentArr = rand() % 10 + 2;

newList->size = sizeOfContentArr;

newList->arr = new int[sizeOfContentArr];

for (int i = 0; i < sizeOfContentArr; i++)

newList->arr[i] = (rand() % 50) \* ((rand()%2)?(-1):(1));

return newList;

}

void addElement(list \* l)

{

while (l->next != NULL)

l = l->next;

list \* newEl = new list;

int sizeOfContentArr = rand() % 10 + 1;

newEl->size = sizeOfContentArr;

newEl->arr = new int[sizeOfContentArr];

for (int i = 0; i < sizeOfContentArr; i++)

newEl->arr[i] = (rand() % 50) \* ((rand() % 2) ? (-1) : (1));

l->next = newEl;

}

void outList(list \* l)

{

for (int i = 0; i < l->size; i++)

cout << l->arr[i] << " ";

cout << endl;

while (l->next != NULL)

{

l = l->next;

for (int i = 0; i < l->size; i++)

cout << l->arr[i] << " ";

cout << endl;

}

}

void deleteNegNumsFromList(list \* l)

{

l->arr = deleteFromArrNeg(l->arr, l->size);

while (l->next != NULL)

{

l = l->next;

l->arr = deleteFromArrNeg(l->arr, l->size);

}

}

int \* deleteFromArrNeg(int\* arr, int &size)

{

int \* tempArr = new int[size];

int tempSize = 0;

for (int i = 0; i < size; i++)

{

if (arr[i] < 0)

continue;

tempArr[tempSize] = arr[i];

tempSize++;

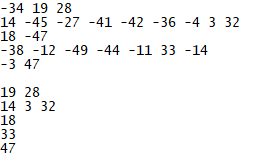
}

size = tempSize;

return tempArr;

}

Результат:



**Задание 3.**

10. Написать программу, содержащую процедуру, которая копирует в список М за каждым вхождением заданного элемента все элемента списка М1.

Код программы:

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

#include <cmath>

using namespace std;

struct list

{

int content;

list \* next = NULL;

};

list \* init(int);

void addElement(list\*, int);

void outList(list\*);

list \* M(list\*, int);

int main()

{

list \* l = init(1);

for (int i = 2; i < 6; i++)

addElement(l, i);

outList(l);

list \* listM = M(l, 1);

outList(listM);

listM = M(l, 2);

outList(listM);

listM = M(l, 3);

outList(listM);

system("pause");

return 0;

}

list \* init(int content)

{

list \* newList = new list;

newList->content = content;

return newList;

}

void addElement(list\* l, int content)

{

list \* newEl = new list;

newEl->content = content;

while (l->next != NULL)

l = l->next;

l->next = newEl;

}

void outList(list\* l)

{

cout << l->content << " ";

while(l->next != NULL)

{

l = l->next;

cout << l->content << " ";

}

cout << endl;

}

list \* M(list\* M1, int minContent)

{

list \* M = new list;

int i = 0;

while (1)

{

if (M1->content > minContent)

{

if (i == 0) M = init(M1->content);

else addElement(M, M1->content);

i++;

}

if (M1->next == NULL) break;

M1 = M1->next;

}

return M;

}

Результат:



**Задание 4.**

2. Написать программу, содержащую процедуру, которая удаляет из списка М N-ый элемент, если такой есть. N задает пользователь.

Код программы:

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

#include <cmath>

using namespace std;

struct list

{

int content;

list \* next = NULL;

};

list \* init(int);

void addElement(list\*, int);

void outList(list\*);

void deleteEl(list\*, int);

list \* deleteFirst(list\*);

int main()

{

list \* l = init(1);

for (int i = 2; i < 6; i++)

addElement(l, i);

outList(l);

int N; cin >> N;

if (N == 1)

l = deleteFirst(l);

else

deleteEl(l, N);

outList(l);

system("pause");

return 0;

}

list \* init(int content)

{

list \* newList = new list;

newList->content = content;

return newList;

}

void addElement(list\* l, int content)

{

list \* newEl = new list;

newEl->content = content;

while (l->next != NULL)

l = l->next;

l->next = newEl;

}

void outList(list\* l)

{

cout << l->content << " ";

while (l->next != NULL)

{

l = l->next;

cout << l->content << " ";

}

cout << endl;

}

void deleteEl(list\* l, int num)

{

int id = num - 1;

for (int i = 1; l->next != NULL; i++)

{

if (i == id)

if (l->next->next == NULL)

{

l->next = NULL;

continue;

}

else l->next = l->next->next;

l = l->next;

}

}

list \* deleteFirst(list\* l)

{

list \* lNew = l->next;

return lNew;

}

Результат:



Вывод: научился работать со стеком и списками.

void deleteEl(list\* l, int num):

