МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

**Списки. Стеки. Очереди**

По лабораторной работе №1

по дисциплине

Технологии программирования

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ [﻿Багиров](http://isu.nntu.ru/?q=node/96) М.Б. (подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Божко М.Е

(подпись) (фамилия, и.,о.)

19-ист-1

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород

2020

**Вариант 7**

Одинокий король долго ходил по бесконечной шахматной доске. Известна последовательность из N его ходов (вверх, вниз, влево, вправо, вверх-влево и т.п.). Написать программу, определяющую побывал ли король дважды на одном и том же поле за минимально возможное при заданном N число вычислений.

Код программы:

#include <iostream>

using namespace std;

struct listItem

{

int x;

int y;

listItem\* next;

listItem\* up;

};

listItem\* createList ( int, int\* );/\*создание списка. Возвращает его адрес, первый парметр кол-во ходов короля, а второй-флаг,

отвечающий за состояние,"не бьл", короля в точке\*/

void printList ( listItem\* );// формальный параметр- адрес списка

listItem\* add ( listItem\*, int, int, int\* );/\* возращаемое значение функции- адрес добавленного эл-та

первый формальный параметр-указатель на текущий элемент, остальные- координаты а последний пар-р - флаг,

отвечающий за состояние "был/не бьл", ф-ия связана с указателем наверх\*/

listItem\* addUp ( listItem\* , int , int , int\* );/\* возращаемое значение функции- адрес добавленного эл-та

первый формальный параметр-указатель на текущий элемент, следующие пар-ры- координаты, а последний пар-р - флаг,

отвечающий за состояние "был", ф-ия связана с указателем по горизонтали \*/

listItem\* newItem ( int, int ); // добавление эл-та, параметры- координаты

int chekNumbers ( int, int ); /\*проверка на неверный ввод числа. формальными парметрами являются

максимальное и минимальное значение диапазона (например,[2;9]) , в котором должно быть введенное число\*/

int main()

{

setlocale ( LC\_ALL, "rus" );

cout << "Как может ходить король:" << endl;

cout << "0 - вверх," << endl;

cout << "1 - вправо - вверх," << endl;

cout << "2 - вправо," << endl;

cout << "3 - вправо - вниз," << endl;

cout << "4 - вниз," << endl;

cout << "5 - влево - вниз" << endl;

cout << "6 - влево," << endl;

cout << "7- влево - вверх" << endl;

cout << "Введите кол-во ходов короля: ";

int n = chekNumbers ( 1, 1000 );

cout << endl << "-------------------------------------------------------" << endl;

int ok = 0;

listItem\* head = createList ( n, &ok );

cout << endl << "-------------------------------------------------------" << endl;

if ( ok == 0 ) cout << "Король не был здесь" << endl;

cout << "Печать координат точек, через которые проходил король" << endl;

printList ( head );

delete head;

return 0;

}

listItem\* newItem ( int x, int y )

{

listItem\* tmp;

tmp = new listItem;

tmp->x = x;

tmp->y = y;

tmp->up = nullptr;

tmp->next = nullptr;

return tmp;

}

int chekNumbers ( int min, int max )

{

char tmp[20];

int i = 0;

int res = 0;

int flag;

while ( 1 )

{

cin >> tmp;

flag = 1;

for ( i = 0; i < strlen ( tmp ); i++)

if ( tmp[i] >='0' && tmp[i] <='9' )

continue;

else

{

cout << "Ошибка!!! Введите значение в диапазоне [" << min << "," << max << "]: " << endl;

flag = 0;

break;

}

if ( flag == 1 )

{

res = atoi ( tmp );

if ( i == strlen ( tmp ) && ( ( res >= min ) && ( res <= max ) ) )

break;

else

cout << "Ошибка!!! Введите значение в диапазоне [" << min << "," << max << "]: " << endl;

}

}

return res;

}

listItem\* addUp ( listItem\* cur, int x, int y, int\* res )

{

listItem\* tmp;

tmp = newItem ( x, y );

if ( y < cur->y )

{

tmp->up = cur;

cur = tmp;

}

else

{

listItem\* prev;

prev = cur;

listItem\* up = cur;

while ( up!= nullptr && y > cur->y )

{

prev = up;

up = up->up;

}

if ( up == nullptr && prev->y == y || up != nullptr && up->y == y ) \*res = 1;

else

{

prev->up = tmp;

if ( up != nullptr ) tmp->up = up;

}

}

return cur;

}

listItem\* add ( listItem\* head, int x, int y, int\* ok )

{

listItem\* cur = nullptr;

listItem\* prev = nullptr;

if ( head == nullptr ) head = newItem ( x, y );

else

{

listItem\* tmp;

tmp = newItem ( x, y );

if ( x < head->x )

{

tmp->next = head;

head = tmp;

}

else

{

cur = head;

prev = nullptr;

while ( cur != nullptr && x > cur->x )

{

prev = cur;

cur = cur->next;

}

int res = 0;

if ( cur != nullptr && cur->x == x )

listItem\* oldCur = cur; //в oldCur хранятся старые координаты.

cur = addUp ( cur, x, y, &res ); //в cur добавились новые координаты.

if ( res == 1 )

{

cout <<endl<< "Король здесь уже был" << endl;

\*ok = 1;

}

else

{

if ( prev == nullptr ) // если head->x==cur->x

{

if ( oldCur != cur ) head = cur; //добавляем в начало.новые координаты

//становятся началом списка

//head==oldCur, head->x=3, а cur->x=2

}

else

{

if ( oldCur != cur ) prev->next = cur; //вставка новых координат между //созданными

//... x=4,y=3, cur->x=5,cur->y=4;, x=5,y=6...

}

cur->next = oldCur->next;

}

}

else if ( cur != nullptr && cur->x > x ) //вставка между эл-ми списка

//head->x=0,head->y=0,tmp->x=2,tmp->y=2,x=3,y=1

{

prev->next = tmp;

if ( cur != nullptr ) tmp->next = cur;

}

else prev->next = tmp; // cur == nullptr и prev->next==nullptr

//prev->x=5, а tmp->x=6

}

}

return head;

}

listItem\* createList ( int n, int\* ok )

{

listItem\* head = nullptr;

listItem\* cur = nullptr;

int x = 0;

int y = 0;

// Добавляем начальную точку

head = add ( head, 0, 0, ok );

while ( n )

{

cout << "Введите значение " ;

int value = chekNumbers ( 0, 7);

switch ( value )

{

case 0:

y++;

break;

case 1:

y++;

x++;

break;

case 2:

x++;

break;

case 3:

y--;

x++;

break;

case 4:

y--;

break;

case 5:

x--;

y--;

break;

case 6:

x--;

break;

case 7:

x--;

y++;

break;

default:

cout << "Ошибка!" << endl;

}

head = add ( head, x, y, ok );

if ( \*ok == 1 )

break;

n--;

}

return head;

}

void printList ( listItem\* head )

{

listItem\* tmp = head;

while ( tmp )

{

cout << "x="<<tmp->x << " y=" << tmp->y;

listItem\* cur = tmp->up;

while ( cur )

{

cout << ", x=" << cur->x << " y=" << cur->y;

cur = cur->up;

}

cout << endl;

tmp = tmp->next;

}

}

Выполнение программы:



