#include <iostream>

using namespace std;

struct Spis2 {

int info;

Spis2\* next, \* prev;

} \*b, \* e, \* t;

struct Spis3 {

int info;

Spis2\* next, \* prev;

} \*x, \* y, \* z;

struct Spis4 {

int info;

Spis2\* next, \* prev;

} \*m, \* n, \* k;

void Del\_All(Spis2\*\*);

void sozd\_Spis2(Spis2\*\*, Spis2\*\*, int);

void dobav(Spis2\*\*, Spis2\*\*, int, int);

void View(Spis2\*\*, Spis2\*\*, int);

void chet(Spis2\*\*);

void nechet(Spis2\*\*);

void sozd\_Spis3(Spis3\*\*, Spis3\*\*, int);

void sozd\_Spis4(Spis4\*\*, Spis4\*\*, int);

int main() {

setlocale(LC\_ALL, "Rus");

int kod, in, n, i;

while (true) {

cout << "1-Создать\n""2-Добавить\n""3-Просмотр\n""4-Освобдить память\n""5-Решение индивидуального задания\n""0-Выход\n";

cin >> kod;

switch (kod) {

case 1:

if (b != NULL) {

cout << "Освободите память!" << endl;

break;

}

cout << "Введите число = "; cin >> in;

sozd\_Spis2(&b, &e, in);

break;

case 2:

if (b == NULL) {

cout << "Создайте список!" << endl;

break;

}

cout << "1-Добавить в начало списка\n""2-Добавить в конец списка \n"; cin >> kod;

cout << "Введите количество = "; cin >> n;

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

cin >> in;

dobav(&b, &e, in, kod);

}

break;

case 3:

if (b == NULL) {

cout << " Список пуст!" << endl;

break;

}

cout << "1-Просмотр с начало списка\n""2-Просмотр с конца списка \n"; cin >> kod;

View(&b, &e, kod);

break;

case 4:

Del\_All(&b);

break;

case 5:

if (b == NULL) {

cout << " Список пуст!" << endl;

break;

}

chet(&b);

nechet(&b);

break;

case 0:

if (b != NULL)

Del\_All(&b);

return 0;

}

}

}

void Del\_All(Spis2\*\* b) {

while (\*b != NULL) {

t = \*b;

\*b = (\*b)->next;

delete t;

}

}

void sozd\_Spis2(Spis2\*\* b, Spis2\*\* e, int in) {

t = new Spis2;

t->info = in;

t->next = t->prev = NULL;

\*b = \*e = t;

}

void sozd\_Spis3(Spis3\*\* x, Spis3\*\* y, int in) {

z = new Spis3;

z->info = in;

z->next = z->prev = NULL;

\*x = \*y = z;

}

void sozd\_Spis4(Spis4\*\* m, Spis4\*\* n, int in) {

k = new Spis4;

k->info = in;

k->next = k->prev = NULL;

\*m = \*n = k;

}

void dobav(Spis2\*\* b, Spis2\*\* e, int in, int kod) {

t = new Spis2;

t->info = in;

if (kod == 1) {

t->prev = NULL;

t->next = (\*b);

(\*b)->prev = t;

(\*b) = t;

}

else {

t->next = NULL;

t->prev = \*e;

(\*e)->next = t;

\*e = t;

}

}

void View(Spis2\*\* b, Spis2\*\* e, int kod) {

t = new Spis2;

if (kod == 1) {

t = (\*b);

while (t != NULL) {

cout << " " << t->info << endl;

t = t->next;

}

}

else {

t = (\*e);

while (t != NULL) {

cout << " " << t->info << endl;

t = t->prev;

}

}

}

void chet(Spis2\*\* b) {

int r;

if (b == NULL) {

cout << "Очередь пуста" << endl;

return;

}

t = (\*b);

cout << "Четная очередь: ";

while (t != NULL) {

if (t->info % 2 == 0) {

r = t->info;

sozd\_Spis3(&x, &y, r);

cout << r << "\t";

t = t->next;

}

else {

t = t->next;

}

}

cout << endl;

}

void nechet(Spis2\*\* b) {

int g;

if (b == NULL) {

cout << "Очередь пуста" << endl;

return;

}

cout << "Нечетная очередь: ";

t = (\*b);

while (t != NULL) {

if (t->info % 2 != 0) {

g = t->info;

sozd\_Spis4(&m, &n, g);

cout << g << "\t";

t = t->next;

}

else {

t = t->next;

}

}

cout << endl;

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Через методы классов

#include<iostream>

using namespace std;

struct Node

{

int x;

Node\* Next;

Node\* Prev;

};

class List

{

Node\* Head;

Node\* Tail;

public:

List(): Head(NULL), Tail(NULL) {};

~List();

void Add(int x);

void Show();

};

List::~List()

{

while(Head)

{

Tail = Head->Next;

delete Head;

Head = Tail;

}

}

void List::Add(int x)

{

Node\* temp = new Node;

temp->Next = NULL;

temp->x = x;

if(Head != NULL)

{

temp->Prev = Tail;

Tail->Next = temp;

Tail = temp;

}

else

{

temp->Prev = NULL;

Head = Tail = temp;

}

}

void List::Show()

{

Node\* temp = Head;

while(temp != NULL)

{

cout << temp->x << " ";

temp = temp->Next;

}

cout <<"\n";

}

int main()

{

int N;

List lst;

cout <<"N = ";

cin >> N;

srand(time(NULL));

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

{

int z = 100 - rand() % 100;

cout <<"\nNumber " << i + 1 <<": " << z;

lst.Add(z);

}

cout <<"\n";

lst.Show();

return 0;

}