#include <iostream>

#include <Windows.h>

using namespace std;

int k = 0;

struct Spis2 {

int info;

Spis2\* next, \* prev;

} \*Begin, \* End, \* t;

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

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

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

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

void Create\_Spis2(Spis2\*\* b, Spis2\*\* e, int in) { // Функция создания первого элемента

t = new Spis2;

t->info = in;

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

\*b = \*e = t;

k++;

}

void Add\_Spis2(int kod, Spis2\*\* b, Spis2\*\* e, int in) { // Функция добавление элемента

t = new Spis2;

t->info = in;

if (kod == 0) {

t->prev = NULL;

t->next = \*b;

(\*b)->prev = t;

\*b = t;

}

else {

t->next = NULL;

t->prev = \*e;

(\*e)->next = t;

\*e = t;

}

k++;

}

void View\_Spis2(int kod, Spis2\* t) { // Функция просмотра элементов

while (t != NULL) {

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

if (kod == 0) t = t->next;

else t = t->prev;

}

}

void Del\_All(Spis2\*\* p) { // Функция удаления элементов

while (\*p != NULL) {

t = \*p;

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

delete t;

}

}

void Task9(Spis2\* Begin) {// Функция удаления элементов списка согласно заданию

int found\_max = 0, found\_min = 0, max = Begin->info, min = Begin->info, n = 0;

bool order = false;

Spis2

\* min\_st = Begin,

\* max\_st = Begin,

\* tmp = Begin;

while (tmp = tmp->next)

{

if (tmp->info > max)

{

max = tmp->info;

max\_st = tmp;

order = false;

found\_max = 0;

}

else

if (tmp->info == max)

{

found\_max++;

}

if (tmp->info < min)

{

min = tmp->info;

min\_st = tmp;

order = true;

found\_min = 0;

}

else

if (tmp->info == min)

{

found\_min++;

}

}

if (found\_min + found\_max > 0)

{

cout << "\n\t\tВ списке присутсвуют несколько максимальных или минимальных чисел\n";

return;

}

if (order)

{

tmp = max\_st;

max\_st = min\_st;

min\_st = tmp;

}

tmp = min\_st->next;

while (tmp != max\_st)

{

t = tmp->next;

delete tmp;

k--;

tmp = t;

n = 1;

}

min\_st->next = max\_st;

max\_st->prev = min\_st;

cout << "\n\t\tМинимальное число в списке = " << min << "\n\t\tМаксимальное число в списке = " << max << endl;

if (n == 1)

cout << "\t\tСписок изменен!\n";

else

cout << "\t\tСписок не изменен:\n\t\tмежду минимальным и максимальным элементами\n\t\tв списке другие элементы отсутсвуют!\n";

return;

}

int main()

{

SetConsoleCP(1251); SetConsoleOutputCP(1251); // Добавление русской локализация

int i = 0, in = 0, kod = 0, kod1 = 0;

char Str[2][10] = { "начало ", "конец " };

while (true) {

cout << "\n\tСоздание - 1"

<< "\n\tДобавление - 2"

<< "\n\tЗадание - 3"

<< "\n\tПросмотр - 4"

<< "\n\tУдаление - 5"

<< "\n\tВыход - 0 : ";

cin >> kod;

switch (kod) {

case 1:

if (Begin != NULL) {

cout << "\n\t\tПамять освобождена!" << endl;

break;

}

cout << "\n\t\tНачальная информация = ";

cin >> in;

Create\_Spis2(&Begin, &End, in);

cout << "\t\tСозданное начало = " << Begin->info << endl;

break;

case 2:

if (!Begin) {

cout << "\n\t\tПамять свободна\n\t\tСоздайте список!" << endl;

break;

}

cout << "\n\t\tИнформация = ";

cin >> in;

cout << "\t\tДобавить в начало - 0, Добавить в конец - 1 : ";

cin >> kod1;

Add\_Spis2(kod1, &Begin, &End, in);

if (kod1 == 0)

t = Begin;

else {

t = End;

kod1 = 1;

}

cout << "\t\tИнформация добавленная в " << Str[kod1] << " = " << t->info << endl; break;

case 3:

if (!Begin) {

cout << "\n\t\tСписок пуст!" << endl;

break;

}

if (!Begin->next) {

cout << "\n\t\tСписок нужно дополнить!" << endl;

break;

}

Task9(Begin);

break;

case 4:

if (!Begin) {

cout << "\n\t\tСписок пуст!" << endl;

break;

}

cout << "\n\t\tКоличество элементов в списке: " << k;

cout << "\n\t\tСмотреть с начала - 0\n"

<< "\t\tСмотреть с конца - 1: ";

cin >> kod1;

if (kod1 == 0) {

t = Begin;

cout << "\n\t\t><> С начала <><" << endl;

}

else {

t = End;

cout << "\n\t\t><> С конца <><" << endl;

}

View\_Spis2(kod1, t);

break;

case 5:

if (!Begin) {

cout << "\n\t\tСписок пуст!" << endl;

break;

}

Del\_All(&Begin);

cout << "\n\t\tПамять освобождена!" << endl;

break;

case 0:

if (Begin != NULL)

Del\_All(&Begin);

return 0;

}

}

}

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

(От Матвея)

#include <iostream>

#include <Windows.h>

using namespace std;

int k = 0;

struct Spis2 {

int info;

Spis2\* next, \* prev;

} \*Begin, \* End, \* t;

void Create\_Spis2(Spis2, Spis2, int);

void Add\_Spis2(int, Spis2, Spis2, int);

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

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

void Create\_Spis2(Spis2 b, Spis2 e, int in) { // Функция создания первого элемента

t = new Spis2;

t->info = in;

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

\*b = \*e = t;

k++;

}

void Add\_Spis2(int kod, Spis2 b, Spis2 e, int in) { // Функция добавление элемента

t = new Spis2;

t->info = in;

if (kod == 0) {

t->prev = NULL;

t->next = \*b;

(\*b)->prev = t;

\*b = t;

}

else {

t->next = NULL;

t->prev = \*e;

(\*e)->next = t;

\*e = t;

}

k++;

}

void View\_Spis2(int kod, Spis2\* t) { // Функция просмотра элементов

while (t != NULL) {

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

if (kod == 0) t = t->next;

else t = t->prev;

}

}

void Del\_All(Spis2\*\* p) { // Функция удаления элементов

while (\*p != NULL) {

t = \*p;

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

delete t;

}

}

void Task9(Spis2\* Begin) {// Функция удаления элементов списка согласно заданию

int found\_max = 0, found\_min = 0, max = Begin->info, min = Begin->info, n = 0;

bool order = false;

Spis2

\* min\_st = Begin,

\* max\_st = Begin,

\* tmp = Begin;

while (tmp = tmp->next)

{

if (tmp->info > max)

{

max = tmp->info;

max\_st = tmp;

order = false;

found\_max = 0;

}

else

if (tmp->info == max)

{

found\_max++;

}

if (tmp->info < min)

{

min = tmp->info;

min\_st = tmp;

order = true;

found\_min = 0;

}

else

if (tmp->info == min)

{

found\_min++;

}

}

if (found\_min + found\_max > 0)

{

cout << "\n\t\tВ списке присутсвуют несколько максимальных или минимальных чисел\n";

return;

}

if (order)

{

tmp = max\_st;

max\_st = min\_st;

min\_st = tmp;

}

tmp = min\_st->next;

while (tmp != max\_st)

{

t = tmp->next;

delete tmp;

k--;

tmp = t;

n = 1;

}

min\_st->next = max\_st;

max\_st->prev = min\_st;

cout << "\n\t\tМинимальное число в списке = " << min << "\n\t\tМаксимальное число в списке = " << max << endl;

if (n == 1)

cout << "\t\tСписок изменен\n";

else

cout << "\t\tСписок не изменен:\n\t\tмежду минимальным и максимальным элементами\n\t\tв списке другие элементы отсутсвуют\n";

return;

}

int Check(char const\* error\_msg) { // Функция проверки вводимых данных пользователем: чисел

char input[100]{};

int ret;

char\* inp = NULL;

do

{

cin >> input;

inp = input;

ret = strtol(input, &inp, 0);

if (input == inp)

cout << error\_msg;

} while (input == inp);

return ret;

}

int main()

{

SetConsoleCP(1251); SetConsoleOutputCP(1251);

int i = 0, in = 0, kod = 0, kod1 = 0;

char Str[2][10] = { "начало ", "конец " };

Strelka Casov, [Сб 04.06.22 23:54]

while (true) {

cout << "\n\tСоздание - 1"

<< "\n\tДобавление - 2"

<< "\n\tЗадание - 3"

<< "\n\tПросмотр - 4"

<< "\n\tУдаление - 5"

<< "\n\tВыход - 0 : ";

kod = Check("\tОшибка: введено не число\n\tВведите число: ");

switch (kod) {

case 1:

if (Begin != NULL) {

cout << "\n\t\tПамять освобождена" << endl;

break;

}

cout << "\n\t\tНачальная информация = ";

in = Check("\t\tОшибка: введено не число\n\t\tВведите число: ");

Create\_Spis2(&Begin, &End, in);

cout << "\t\tСозданное начало = " << Begin->info << endl;

break;

case 2:

if (!Begin) {

cout << "\n\t\tПамять свободна\n\t\tСоздайте список" << endl;

break;

}

cout << "\n\t\tИнформация = ";

in = Check("\t\tОшибка: введено не число\n\t\tВведите число: ");

cout << "\t\tДобавить в начало - 0, Добавить в конец - 1 : ";

kod1 = Check("\t\tОшибка: введено не число\n\t\tВведите число: ");

Add\_Spis2(kod1, &Begin, &End, in);

if (kod1 == 0)

t = Begin;

else {

t = End;

kod1 = 1;

}

cout << "\t\tИнформация добавленная в " << Str[kod1] << " = " << t->info << endl; break;

case 3:

if (!Begin) {

cout << "\n\t\tСписок пуст" << endl;

break;

}

Task9(Begin);

break;

case 4:

if (!Begin) {

cout << "\n\t\tСписок пуст" << endl;

break;

}

cout << "\n\t\tКоличество элементов в списке: " << k;

cout << "\n\t\tСмотреть с начала - 0\n"

<< "\t\tСмотреть с конца - 1: ";

kod1 = Check("\t\tОшибка: введено не число\n\t\tВведите число: ");

if (kod1 == 0) {

t = Begin;

cout << "\n\t\tС начала:" << endl;

}

else {

t = End;

cout << "\n\t\tС конца:" << endl;

}

View\_Spis2(kod1, t);

break;

case 5:

if (!Begin) {

cout << "\n\t\tСписок пуст" << endl;

break;

}

Del\_All(&Begin);

cout << "\n\t\tПамять освобождена" << endl;

break;

case 0:

if (Begin != NULL)

Del\_All(&Begin);

return 0;

}

}

}

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

(От Ильи)

#include <iostream>

using namespace std;

struct List {

int data;

List \*next;

List \*prev;

};

List\* push(List\* head, int newData) {

List\* ptr = new List;

ptr->data = newData;

ptr->next = head;

return ptr;

}

void view(List\* head) {

List\* ptr = head;

while (ptr != NULL) {

cout << ptr->data << " ";

ptr = ptr->next;

}

cout << endl;

}

int max\_pos\_f(List\* head) {

List\* ptr = head;

int max = ptr->data;

int pos = 0;

int i = 0;

while (ptr != NULL) {

if (ptr->data > max) {

max = ptr->data;

pos = i;

}

i++;

ptr = ptr->next;

}

return pos;

}

int min\_pos\_f(List\* head) {

List\* ptr = head;

int min = ptr->data;

int pos = 0;

int i = 0;

while (ptr != NULL) {

if (ptr->data < min) {

min = ptr->data;

pos = i;

}

i++;

ptr = ptr->next;

}

return pos;

}

List\* pop(List\* head) {

head = head->next;

return head;

}

void del(List\*\* head) {

List\* ptr;

while (\*head != NULL) {

ptr = \*head;

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

delete ptr;

}

}

void sort1(List\* head) {

List\* ptr = NULL, \* ptr1;

int tmp;

do {

for (ptr1 = head; ptr1->next != ptr; ptr1 = ptr1->next) {

if (ptr1->data > ptr1->next->data) {

tmp = ptr1->data;

ptr1->data = ptr1->next->data;

ptr1->next->data = tmp;

}

}

ptr = ptr1;

} while (head->next != ptr);

}

void sort2(List\*\* head) {

List\* ptr = NULL, \* ptr1, \* tmp;

if ((\*head)->next->next == NULL) return;

do {

for (ptr1 = \*head; ptr1->next->next != ptr; ptr1 = ptr1->next)

if (ptr1->next->data > ptr1->next->next->data) {

tmp = ptr1->next->next;

ptr1->next->next = tmp->next;

tmp->next = ptr1->next;

ptr1->next = tmp;

}

ptr = ptr1->next;

} while ((\*head)->next->next != ptr);

}

int main() {

int n = 0;

List\* head = NULL;

cout << "Enter size of List:" << endl;

cin >> n;

cout << "Fill the List:" << endl;

int newData = 0;

for (int i = 0; i < n; i++) {

cin >> newData;

head = push(head, newData);

}

cout << "List: ";

view(head);

int k = 0;

cout << "Enter 1 - to sort with the first method, 2 - to sort with the second method, 3 - to

delete numbers between min and max values" << endl;

cin >> k;

switch (k)

{

case(1):

sort2(&head);

break;

case(2):

sort1(head);

break;

case(3):

int max\_pos = max\_pos\_f(head);

int min\_pos = min\_pos\_f(head);

List\* ptr\_min, \* ptr\_max;

if (max\_pos == min\_pos) break;

if (max\_pos > min\_pos) {

cout << "Elements between:" << max\_pos - min\_pos - 1 << endl;

ptr\_max = head;

for (int i = 0; i < max\_pos; i++) {

ptr\_max = ptr\_max->next;

}

for (int i = 0; i <= min\_pos; i++) {

cout << head->data << " ";

head = head->next;

}

view(ptr\_max);

}

else {

cout << max\_pos << " " << min\_pos << endl;

cout << "Elements between:" << -max\_pos + min\_pos - 1 << endl;

cout << "New List:";

ptr\_min = head;

for (int i = 0; i < min\_pos; i++) {

ptr\_min = ptr\_min->next;

}

for (int i = 0; i <= max\_pos; i++) {

cout << head->data << " ";

head = head->next;

}

view(ptr\_min);

}

break;

}

return 0;

}