|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 6 |

**Название:**

Динамические структуры данных. Списки C++.

**Дисциплина:** Объектно-ориентированное программирование

Вариант 16.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-21Б |  |  | Т.Е.Старжевский |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | О.А.Веселовская |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Лабораторная работа 6. Динамические структуры данных. Списки**

С клавиатуры вводится символьная строка, содержащая последовательность вещественных чисел x1, x2, x3... xn (n>2) в символьном представлении. Числа разделены пробелами. Составить из чисел последовательности двусвязный список, предварительно преобразовав числа в вещественное представление. Сформировать новый список, состоящий из чисел: **min**(x1,xn); **min**(x2,xn-1); **min**(x3,xn-2), ..., **min**(xn,x1). Вывести на печать оба списка.

**Блок схема:**



**Код программы:**

#include <iostream>

using namespace std;

char\* s1, stroka[255];

struct spisok

{

float num;

spisok\* next;

spisok\* prev;

};

spisok \*dvus, \*q, \*dvus\_new, \*q2, \*r,\*r2;

float min(float a, float b)

{

if (a > b)

return b;

else

return a;

}

void show (spisok \*a)

{

int i = 0;

while (a != nullptr)

{

if (i % 5 == 0)

{

cout << endl << ++i << ") " << a->num << "\t\t";

a = a->next;

}

else

{

cout << ++i << ") " << a->num << "\t\t";

a = a->next;

}

}

if (i % 5 == 4)

cout << endl;

}

int main()

{

gets\_s(stroka);

if (strcmp(stroka, "") != 0)

{

dvus = new spisok;

dvus->num = strtod(stroka, &s1);

dvus->next = nullptr;

dvus->prev = nullptr;

q = dvus;

while (strcmp(s1, "") != 0)

{

q->next = new spisok;

q->next->prev = q;

q = q->next;

q->next = nullptr;

q->num = strtod(s1, &s1);

}

cout << endl;

//Working

r = q; r2 = dvus;

dvus\_new = new spisok;

dvus\_new->prev = nullptr;

dvus\_new->next = nullptr;

dvus\_new->num = min(r2->num, r->num);

r2 = r2->next;

r = r->prev;

q2 = dvus\_new;

while (r != nullptr)

{

q2->next = new spisok;

q2->next->prev = q2;

q2 = q2->next;

q2->next = nullptr;

q2->num = min(r2->num, r->num);

r2 = r2->next;

r = r->prev;

}

r = dvus; show(r);

cout << endl;

r = dvus\_new; show(r);

//Delete

q = q->prev;

while (q != nullptr)

{

delete q->next;

q = q->prev;

}

delete dvus;

q2 = q2->prev;

while (q2 != nullptr)

{

delete q2->next;

q2 = q2->prev;

}

delete dvus\_new;

}

return 0;

}

**Тестирование:**

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Полученный результат |
|  | Вывод отсортированного по минимальным крайним элементам списка |  |

При заданных значениях программа работает корректно.

**Вывод:** Научился работать с динамической памятью и списками на языке C++.