МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Череповецкий государственный университет»

**Лабораторная работа № 3**

**«Полный перебор»**

**Выполнил:**

**студент гр.** 1ИВТпб-01-21оп

Климов А.Г. **Проверил: преподаватель**

Пышницкий К.М. **Отметка о зачете:**

Череповец

2017 год

**Ход работы**

**Задание 5.** Решить методом полного p-ичного перебора следующую задачу:

5. Гвозди и веревка

В доску в один ряд вбито N гвоздей (N<=20). Расстояния между соседними гвоздями - натуральные числа, записанные в текстовый файл по одному в каждой строке. Имеется веревка длины L. Веревку требуется разрезать на несколько частей так, чтобы каждой частью можно было связать какие-либо два соседних гвоздя и никакие два гвоздя не были связаны более одного раза. На привязывание веревки к гвоздю уходит Р единиц длины. Числа P и L вводятся с клавиатуры. Вывести в текстовый файл все возможные варианты соединения гвоздей частями веревки. Каждый вариант записать в виде строки, состоящей из чисел 1, 2, 3, :, N (номера гвоздей), между которыми стоят пробелы (гвозди не связаны) или знаки минус (гвозди связаны). Например, при N=4, L=19, P=1 и расстояниях между гвоздями 5, 10, 17 выходной файл может иметь вид:

1-2-3 4

1 2 3-4

**Логическое проектирование:**

1. Проверяем на наличие чисел (расстояний между гвоздями) и подсчитываем, сколько их в файле;
2. Исходя из количества чисел (расстояний между гвоздями), находим количество гвоздей.
3. Записываем числа (расстояния между гвоздями) в динамический массив (если числа есть в файле);
4. Проверяем, если расстояний между гвоздями в файле нет или больше и гвоздей больше 20, то программа прекращает работу, иначе программа продолжает работу;
5. Формируем динамический бинарный массив из нулей для полного перебора;
6. Ввод P и L.
7. Создаём динамический массив для вывода результата.
8. Выполняем перебор.
9. Если в бинарном массиве 0, то гвозди не связаны (записываем связь между гвоздями – пробел в массив для вывода результата). Если в бинарном массиве 1, то гвозди связаны (находим длину использованной верёвки).
10. Проверяем, вся ли верёвка использована. Если условие выполняется, то записываем результат.
11. Повторяем п.9 и п.10 пока не выполним перебор.
12. Удаление динамических массивов.

**Текст программы:**

**а)**

#include "stdafx.h"

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int \*bin, n;

int \*num;

char \*temp;

int x;

int xx = 1;

int count\_num = 0;

ifstream f("input.txt");

while (!f.eof()) {

if (!(f >> x)) break;

count\_num++;

}

f.close();

cout << "Чисел в файле: " << count\_num << endl;

num = new int[count\_num + 1];

num[0] = 0;

f.open("input.txt");

while (!f.eof()) {

if (!(f >> x)) break;

num[xx] = x; xx++;

}

f.close();

n = count\_num + 1;

if (n > 20 || count\_num <= 0) { return 0; }

else {

cout << "Вывод сформированного массива из чисел с файла: ";

for (int i = 0; i <= count\_num; i++) {

cout << num[i] << " ";

}

cout << endl;

bin = new int[n];

cout << "Вывод сформированного бинарного массива: ";

for (int i = 0; i <= n - 1; i++) {

bin[i] = 0; cout << bin[i] << " ";

} //заполнение бинарного массива нулями

cout << endl;

int P;

int L;

int l1, l2;

cout << "N=" << n << endl;

cout << "P=";

cin >> P;

cout << "L=";

cin >> L;

cout << "Результат:" << endl;

ofstream f1("output.txt");

temp = new char[n];

temp[0] = ' ';

while (bin[0] == 0) {

int k = 1;

int tmpL = 0;

for (int i = 1; i <= n - 1; i++)

{

if (bin[i] == 0)temp[i] = ' ';

else { tmpL += num[i] + 2 \* P; temp[i] = '-'; }

}

if (tmpL == L) { for (int i = 1; i <= n - 1; i++) { cout << k << temp[i]; f1 << k << temp[i]; k++; }; f1 << k << endl; cout << k << endl; }

int j = n - 1;

while (bin[j] == 1)

{

bin[j] = 0; j--;

}

bin[j] = 1;

k = 1;

}

f1.close();

delete[] bin;

delete[] num;

}

delete[]temp;

cout << endl;

system("pause");

return 0;

}

**Тестирование:**

1)Ввод “input.txt”:

5

10

17

N=4

P=1

L=19

Вывод “output.txt”:

1 2 3-4

1-2-3 4

2)Ввод “input.txt”:

5

10

15

20

25

N=6

P=0

L=50

Вывод “output.txt”:

1 2-3-4 5-6

1-2 3 4-5-6

1-2-3-4-5 6

3) Ввод “input.txt”:

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

N=21

P=1

L=50

Вывод:

Программа при N>20 закрывается.

4) Ввод “input.txt”:

5

10

15

20

25

N=6

P=0

L=75

Вывод “output.txt”:

1-2-3-4-5-6