**2 ЛАБОРАТОРНАЯ РАБОТА №2**

**«Исследование способов структурного тестирования**

**программного обеспечения»**

**2.1 Цель работы**

Исследовать основные подходы к структурному тестированию программного обеспечения. Приобрести практические навыки построения графа потоков управления и определения независимых ветвей программы.

**2.2 Вариант задания**

Задача 1. Дана целочисленная прямоугольная матрица. Определить номер строки, в которой находится самая длинная серия нулевых элементов.

Задача 2. Дана строка. Преобразовать строку, заменив в ней все вопросительные знаки амперсантами.

Задача 3. Программа, которая считывает текст из файла и выводит на экран только строки, содержащие трехзначные числа.

**2.3 Ход выполнения работы**

2.3.1 Программы, написанные в предыдущей лабораторной работе, были исследованы с помощью структурного тестирования. Код первой программы представлен в листинге 2.2.

Листинг 2.1 – Текст первой программы

#include <iostream>

using namespace std;

int cntStr(int \*\*a, int n, int m);

int main() {

int n, m;

int \*\*a;

cout << "Enter N\*M: ";

cin >> n >> m;

a = new int \*[n];

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

a[i] = new int[m];

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

for (int j = 0; j < m; j++) {

cout << "[" << i << "][" << j << "] = ";

cin >> a[i][j];

}

}

cout << cntStr(a, n, m);

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

delete[] a[i];

delete[] a;

return 0;

}

int cntStr(int \*\*a, int n, int m) {

int max = 0, maxCur = 0, maxTmp = 0, maxStr = -1;

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

maxTmp = 0;

for (int j = 0; j < m; j++) {

if (a[i][j] == 0)

maxTmp++;

else {

if (maxTmp > maxCur) {

maxCur = maxTmp;

}

maxTmp = 0;

}

}

if (maxTmp > maxCur) {

maxCur = maxTmp;

}

if (maxCur > max) {

max = maxCur;

maxStr = i;

}

maxCur = 0;

}

if (maxStr == -1) return 0;

return maxStr + 1;

}

Для этой программы был построен граф потоков управления, который показан на рисунке 2.1. Цикломатическое число для этого графа:

С(G) = 17-13+2=6.

Следует вывод, что граф имеет 6 независимых ветвей.

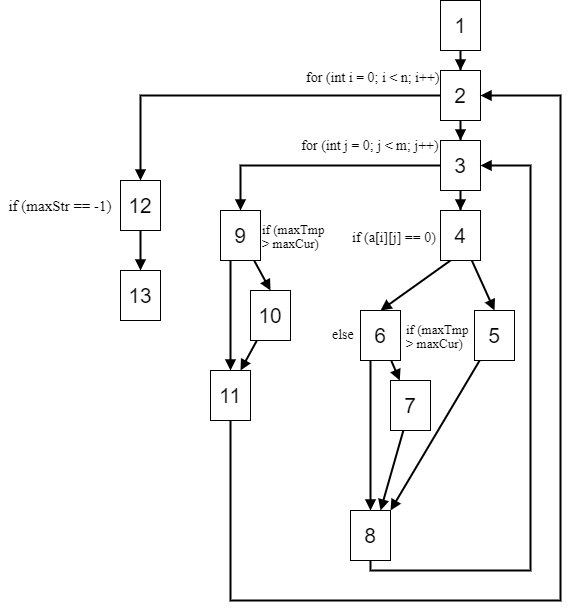


Рисунок 2.1 – Граф потоков управления первой программы

Были найдены все 7 независимых ветвей этой программы:

1. 1, 2, 3, 4, 5, 8, 3, 9, 10, 11, 2, 12, 13
2. 1, 2, 3, 4, 6, 7, 8, 3, 9, 10, 11, 2, 12, 13
3. 1, 2, 3, 4, 6, 7, 8, 3, 9, 11, 2, 12, 13
4. 1, 2, 3, 4, 6, 8, 3, 9, 10, 11, 2, 12, 13
5. 1, 2, 3, 4, 6, 7, 8, 3, 9, 11, 2, 12, 13
6. 1, 2, 12, 13

2.3.2 Для следующей программы были проделаны те же действия. Код программы представлен в листинге 2.2.

Листинг 2.2 – Текст второй программы

#include <iostream>

#include <string>

using namespace std;

string replace(string str);

int main(){

string str;

cout<<"Enter string: ";

cin>>str;

cout<<replace(str);

return 0;

}

string replace(string str){

for(int i = 0; i < str.length(); i++){

if(str[i] == '?')

str[i]='&';

}

return str;

}

Для второй программы был построен граф потоков управления, который показан на рисунке 2.2. Цикломатическое число для этого графа:

С(G) = 7-6+2=3.

Следует вывод, что граф имеет 7 независимых ветвей.

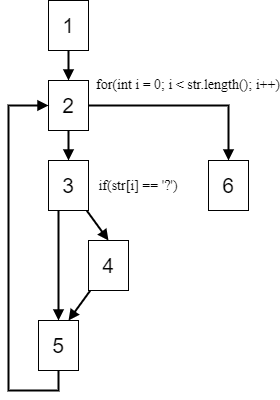


Рисунок 2.2 – Граф потоков управления второй программы

Были найдены все 3 независимых ветви этой программы:

1. 1, 2, 6
2. 1, 2, 3, 4, 5, 2, 6
3. 1, 2, 3, 5, 2, 6

2.3.3 Последней была исследована программа, которая считывает текст из файла и выводит на экран только строки, содержащие трехзначные числа. Код программы представлен в листинге 2.3.

Листинг 2.3 – Текст третьей программы

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main(){

string str, line;

cout<<"Enter file name: ";

cin>>str;

string filename = str + ".txt";

ifstream fin(filename);

if(!fin){

cout<<"Error opening file";

return 1;

}

bool lineFlag = false;

while(getline(fin, line)){

for(int i = 0; i<line.length()-2; i++){

if (isdigit(line[i]) && isdigit(line[i + 1]) && isdigit(line[i + 2])) {

if ((i == 0 || !isdigit(line[i - 1])) && (!isdigit(line[i + 3]) || line.length() == i + 3)){

cout << line << endl;

lineFlag = true;

break;

}

}

}

}

if(!lineFlag) cout<<"No suitable lines";

return 0;

}

Для третьей программы был построен граф потоков управления, который показан на рисунке 2.3. Цикломатическое число для этого графа:

С(G) = 17-12+2=7.

Следует вывод, что граф имеет 4 независимых ветвей.

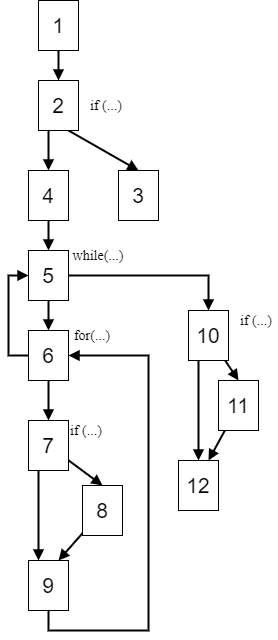


Рисунок 2.3 – Граф потоков управления третьей программы

Были найдены все 5 независимых ветвей этой программы:

1. 1, 2, 3
2. 1, 2, 4, 5, 6, 7, 8, 9, 6, 5, 10, 11, 12
3. 1, 2, 4, 5, 6, 7, 8, 9, 6, 5, 10, 12
4. 1, 2, 4, 5, 6, 7, 9, 6, 5, 10, 11, 12
5. 1, 2, 4, 5, 6, 7, 9, 6, 5, 10, 12
6. 1, 2, 4, 5, 6, 5, 10, 11, 12
7. 1, 2, 4, 5, 6, 5, 10, 12

**Выводы**

В ходе лабораторной работы были исследованы основные подходы к структурному тестированию программного обеспечения. Также были приобретены практические навыки построения графа потоков управления и определения независимых ветвей программы. В конце выполнения лабораторной работы был написан отчет.