МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

кафедра «Вычислительная техника»

**Лабораторная работа №3**

по дисциплине: «Теория автоматов».

Вариант №12

Выполнил:

студент гр. ИВТАПбд-21

Кондратьев П.С.

Проверил:

преподаватель кафедры ВТ

Лылова А.В.

г. Ульяновск, 2018

**Техническое Задание:**

Требуется выполнить программную реализацию автомата, вычисляющего значения арифметических выражений. Автомат выполняется в виде класса, содержащего метод, принимающий на вход выражение, а на выходе выдающий булевское значение, показывающее, является ли выражение корректным. Другим методом реализуемого класса является метод получения результата вычисления.

Выражение в класс передается как строка. Строка может содержать пробелы между операндами, операциями и скобками.

Вариант Задания:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 12 | С плавающей точкой | -32768 | 32767 | Мили |

**Исходный код программы:**

#include <iostream>

#include <string>

#include <stack>

#include <sstream>

using namespace std;

class mat {

public:

string s;

void check(string str) {

s = str;

string st, error;

int k = 0, parenthesis = 0;

for (int i = 0; i <= s.size(); i++) {

if (s[i] == '(') {

parenthesis++;

k++;

}

if (s[i] == ')') {

parenthesis--;

}

if (s[i] == '+' || s[i] == '-' || s[i] == '/' || s[i] == '\*' || s[i] == '\0') {

st = s.substr(k, i - k);

if (st == "")

continue;

k = i + 1;

if (st != "(" && st.find('.') == std::string::npos) {

error += "is not real value\n";

}

if (strtol(st.c\_str(), NULL, 10) > 32767 || strtol(st.c\_str(), NULL, 10) < -32768) {

error += "the value of the number overflow\n";

}

st.clear();

}

}

if (parenthesis != 0)

error += "syntax error";

if (error.empty()) {

cout << "|True|\nNot error\n";

} else {

cout << "|False|\nError:\n" << error;

}

}

void RPN(string str) {

s = str;

stack<pair<char, double>> p;

string res;

int k = 0, temp = 0;

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

if (s[i] == ' ')

i++;

if (isdigit(s[i]) || s[i] == '.') {

do {

res += s[i];

i++;

} while (isdigit(s[i]) || s[i] == '.');

res += " ";

i--;

}

else if (s[i] == '\*' || s[i] == '/')

k = 3;

else if (s[i] == '+' || s[i] == '-')

k = 2;

else if (s[i] == '(' || s[i] == ')')

k = 1;

if (s[i] == '(') {

p.push(make\_pair(s[i], k));

k = 0;

}

if ((s[i] == '(') && (s[i + 1] == '-')) {

res += "0.0 ";

}

if (s[i] == ')') {

while (p.top().first != '(') {

res += p.top().first;

res += " ";

p.pop();

}

p.pop();

k = 0;

}

if (k > 0 && (p.empty() || p.top().second < k)) {

p.push(make\_pair(s[i], k));

k = 0;

}

if (k > 0 && p.top().second >= k) {

while (p.top().second >= k) {

res += " ";

res += p.top().first;

res += " ";

p.pop();

if (p.empty())

break;

}

if (k > 0 && (p.empty() || p.top().second < k)) {

p.push(make\_pair(s[i], k));

k = 0;

}

}

}

while (!p.empty()) {

res += " ";

res += p.top().first;

p.pop();

}

s = res;

}

void RP() {

stack<double> a;

double p1, p2;

istringstream is;

is.str(s);

while (is >> s) {

if (s == "+") {

p1 = a.top();

a.pop();

p2 = a.top();

a.pop();

a.push(p1 + p2);

}

else if (s == "\*") {

p1 = a.top();

a.pop();

p2 = a.top();

a.pop();

a.push(p1 \* p2);

}

else if (s == "-") {

p1 = a.top();

a.pop();

p2 = a.top();

a.pop();

a.push(p2 - p1);

}

else if (s == "/") {

p1 = a.top();

a.pop();

p2 = a.top();

a.pop();

a.push(p2 / p1);

}

else a.push(stod(s));

}

cout << a.top();

}

};

int main() {

/\*

Test

2.2 \* (3.1 - 1.1) / 3.3 - 22.3 \* (123.123 - 111.1)

2.2 \* (17.1 + (-2.1)) / 3.3 - 22.3 \* (123.123 - 111.1)

\*/

string s;

getline(cin, s);

mat d;

d.check(s);

d.RPN(s);

cout << d.s << "\n";

d.RP();

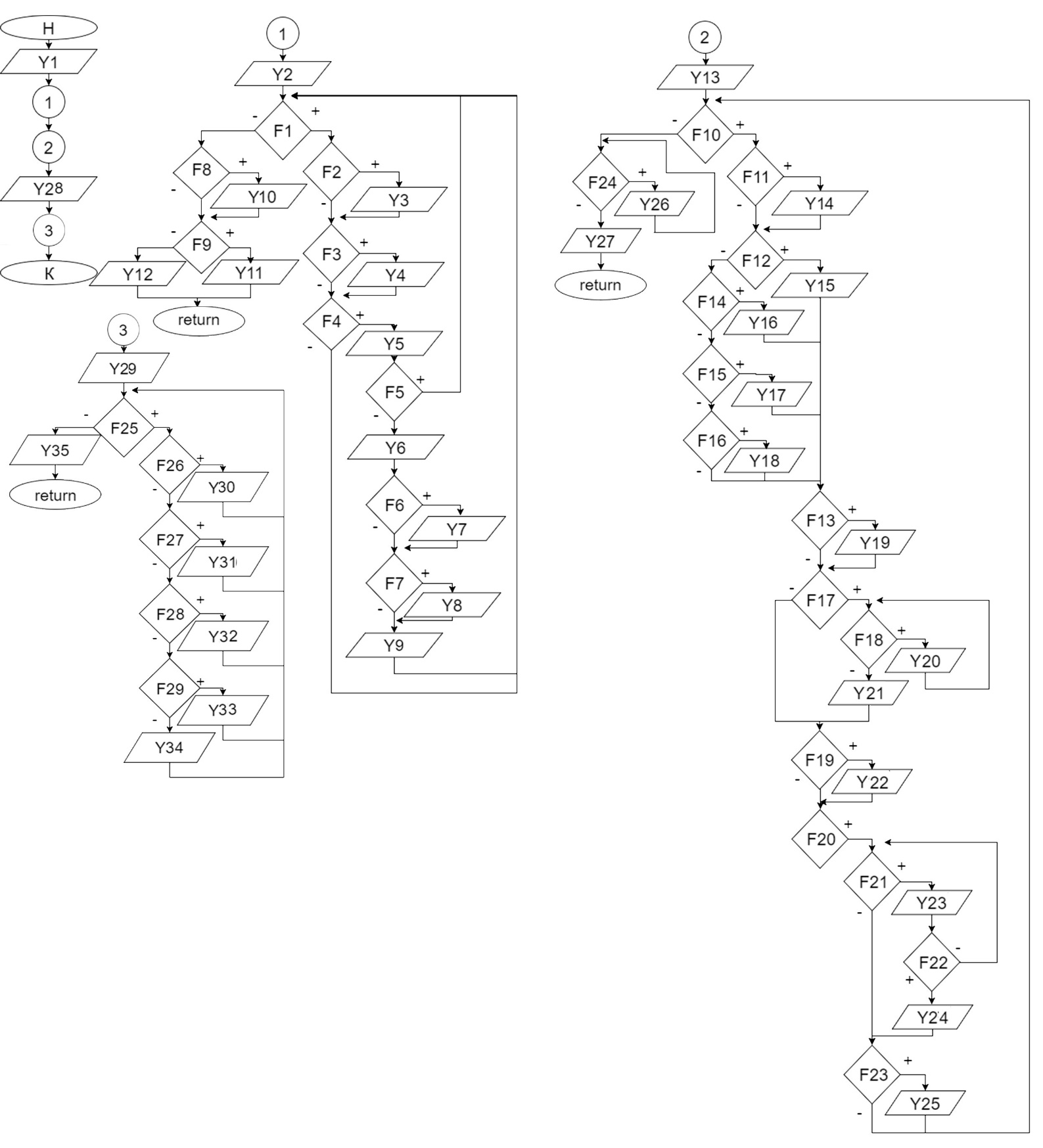
}

|  |  |
| --- | --- |
| **Код ошибки** | **Ошибка** |
| **#**042 | Error: is not real value |
| #420 | Error: the value of the number overflow |
| #024 | Error: syntax error |

**Результат работы программы:**

1. 2.2 + 2.2 => 2.2 2.2 + => 4.4
2. 2.2 \* (3.1 - 1.1) / 3.3 - 22.3 \* (123.123 - 111.1) => 2.2 3.1 1.1 - \* 3.3 / 22.3 123.123 111.1 - \* - => -266.78
3. 2.2 \* (17.1 + (-2.1)) / 3.3 - 22.3 \* (123.123 - 111.1) => 2.2 17.1 0.0 2.1 - + \* 3.3 / 22.3 123.123 111.1 - \* - => -258.113

**Блок-схема и граф:**

****

**Кодирование:**

|  |  |
| --- | --- |
| Y | Соответствующие функции |
| Y1 | string s;  getline(cin, s);  mat d; |
| Y2 | s = str;  string st, error;  int k = 0, parenthesis = 0; |
| Y3 | parenthesis++;  k++; |
| Y4 | parenthesis--; |
| Y5 | st = s.substr(k, i - k); |
| Y6 | k = i + 1; |
| Y7 | error += "is not real value\n"; |
| Y8 | error += "the value of the number overflow\n"; |
| Y9 | st.clear(); |
| Y10 | error += "syntax error"; |
| Y11 | cout << "|True|\nNot error\n"; |
| Y12 | cout << "|False|\nError:\n" << error; |
| Y13 | s = str;  stack<pair<char, double>> p;  string res;  int k = 0, temp = 0; |
| Y14 | i++; |
| Y15 | k = 3; |
| Y16 | k = 2; |
| Y17 | k = 1; |
| Y18 | p.push(make\_pair(s[i], k));  k = 0; |
| Y19 | res += "0.0 "; |
| Y20 | res += p.top().first;  res += " ";  p.pop(); |
| Y21 | p.pop();  k = 0; |
| Y22 | p.push(make\_pair(s[i], k));  k = 0; |
| Y23 | res += " ";  res += p.top().first;  res += " ";  p.pop(); |
| Y24 | break; |
| Y25 | p.push(make\_pair(s[i], k));  k = 0; |
| Y26 | res += " ";  res += p.top().first;  p.pop(); |
| Y27 | s = res; |
| Y28 | cout << d.s << "\n"; |
| Y29 | stack<double> a;  double p1, p2;  istringstream is;  is.str(s); |
| Y30 | p1 = a.top();  a.pop();  p2 = a.top();  a.pop();  a.push(p1 + p2); |
| Y31 | p1 = a.top();  a.pop();  p2 = a.top();  a.pop();  a.push(p1 \* p2); |
| Y32 | p1 = a.top();  a.pop();  p2 = a.top();  a.pop();  a.push(p2 - p1); |
| Y33 | p1 = a.top();  a.pop();  p2 = a.top();  a.pop();  a.push(p2 / p1); |
| Y34 | a.push(stod(s)); |
| Y35 | cout << a.top(); |
| F1 | i <= s.size() |
| F2 | s[i] == '(' |
| F3 | s[i] == ')' |
| F4 | s[i] == '+' || s[i] == '-' || s[i] == '/' || s[i] == '\*' || s[i] == '\0' |
| F5 | st == "" |
| F6 | st != "(" && st.find('.') == std::string::npos |
| F7 | strtol(st.c\_str(), NULL, 10) > 32767 || strtol(st.c\_str(), NULL, 10) < -32768 |
| F8 | parenthesis != 0 |
| F9 | error.empty() |
| F10 | i < s.length() |
| F11 | s[i] == ' ' |
| F12 | isdigit(s[i]) || s[i] == '.' |
| F13 | s[i] == '(' |
| F14 | s[i] == '\*' || s[i] == '/' |
| F15 | s[i] == '+' || s[i] == '-' |
| F16 | s[i] == '(' || s[i] == ')' |
| F17 | (s[i] == '(') && (s[i + 1] == '-') |
| F18 | s[i] == ')' |
| F19 | p.top().first != '(' |
| F20 | k > 0 && (p.empty() || p.top().second < k) |
| F21 | k > 0 && p.top().second >= k |
| F22 | p.top().second >= k |
| F23 | p.empty() |
| F24 | k > 0 && (p.empty() || p.top().second < k) |
| F25 | is >> s |
| F26 | s == "+" |
| F27 | s == "\*" |
| F28 | s == "-" |
| F29 | (s == "/" |