

МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение высшего образования*

***«МИРЭА – Российский технологический университет»***

**РТУ МИРЭА**

Отчет по выполнению практического задания № 7

**Тема**: **Линейные списки: стека и очередь (реализация, применение)**

Дисциплина: Структуры и алгоритмы обработки данных

|  |  |
| --- | --- |
| Выполнил студент | Дмитриев П. В. |
|  | Фамилия И.О. |
| Группа | ИКБО-33-21 |
|  | Номер группы |

Москва 2022

Вариант № 7

Тема. Использование линейных структур данных стека и очереди в алгоритмах Использование стека и очереди в алгоритмах преобразования инфиксной записи арифметических выражений в польскую запись и вычисление значений выражений

Цель.

* Получить знания и навыки по реализации структуры стек и очередь
* Получить умения и навыки по выполнению операций на структурах стек и очередь
* Получить знания, умения по представлению арифметических выражения в польской записи

ЗАДАНИЕ 1

Провести преобразование инфиксной записи выражения (столбец 1 таблицы вариантов) в постфиксную нотацию, расписывая процесс по шагам.

1. **Разработка программы задачи 1**
   1. **Постановка задачи**

Дано. Дана строка, содержащая инфиксную запись выражения

Результат. Вывод постфиксной нотации выражения

* 1. **Описание модели решения**

Дана строка, содержащая инфиксную запись выражения. Вызывается функция postn(in), которой передается строка в виде параметра. Внутри функции postn поэлементно перебирается строка с помощью цикла for. В цикле происходит проверка символов, если символ является буквой, то он записывается в новую строку post. Если символ является оператором, то сначала проверяется, присутствует ли на конце стека предыдущий оператор с высшим приоритетом, если нет, то текущий оператор добавляется в конец стека, если да, то в строку post добавляется предыдущий оператор и сразу же удаляется из стека, а на его место встает текущий оператор. Если символ является “(“, то он добавляется в стек. Если символ является “)”, то последний элемент стека переписывается в строку post и затем удаляется, пока последний элемент не стал “(“, после удаляется сама “(“.

Функция Operator(char x) проверяет элемент на его принадлежность к оператору.

Функция Priority(char x) возвращает приоритет операторам.

**1.3 Декомпозиция:**

1. Проверка символа на оператора

Предусловие. x – символ из введенной строки

Постусловие. Возврат 1 или 0

bool Operator(char x)

1. Возврат приоритета оператору

Предусловие. x – символ из введенной строки

Постусловие. Возврат 1 или 2

int Priority(char x)

1. Возврат выражения в постфиксной нотации

Предусловие. in – введенная строка

Постусловие. Возврат строки

string postn(string in)

**1.4 Код программы:**

#include <iostream>

#include <stack>

#include <string>

using namespace std;

bool Operator(char x) {

return x == '\*' or x == '/' or

x == '+' or x == '-';

}

int Priority(char x) {

if (x == '+' or x == '-') {

return 1;

}

if (x == '\*' or x == '/') {

return 2;

}

return 0;

}

string postn(string in) {

string post;

stack<char> stek;

for (char element : in) {

if (isalpha(element)) {

post += element;

}

if (Operator(element)) {

if (Priority(stek.top()) < Priority(element)) {

stek.push(element);

}

else {

while (!stek.empty() and stek.top() != '(' and !(Priority(stek.top()) < Priority(element))) {

post += stek.top();

stek.pop();

}

stek.push(element);

}

}

if (element == '(') {

stek.push(element);

}

if (element == ')') {

while (!stek.empty() and stek.top() != '(') {

post += stek.top();

stek.pop();

}

if (stek.top() == '(') {

stek.pop();

}

}

}

return post;

}

int main() {

setlocale(LC\_ALL, "RUS");

string in = "((a\*(d-e)+(f+g)/c)+(k\*l-s/m))";

cout << "Инфиксная нотация: " << in << endl;

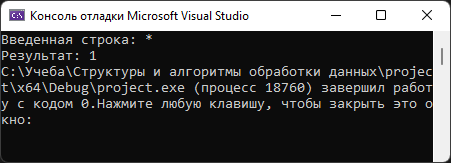
string post = postn(in);

cout << "Выражение, записанное в постфиксной нотации: " << post << endl;

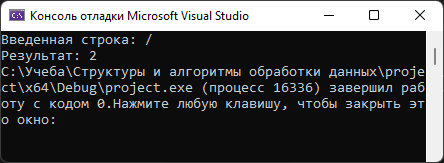
}

* 1. **Скриншоты:**

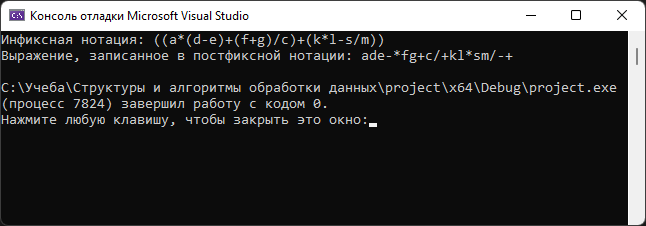
1. Тестирование функции декомпозиции: проверка символа на оператора



1. Тестирование функции декомпозиции: возврат приоритета оператору



1. Тестирование функции декомпозиции: возврат выражения в постфиксной нотации



ЗАДАНИЕ 2

Представить инфиксную нотацию выражения (столбец 2 таблицы вариантов) (идентификаторы одно символьные) с расстановкой скобок, расписывая процесс по шагам.

1. **Разработка программы задачи 2**
   1. **Постановка задачи**

Дано. Дана строка, содержащая постфиксную запись выражения

Результат. Вывод инфиксной нотации выражения

* 1. **Описание модели решения**

Вызывается функция in(post) с параметром строки. Внутри нее в цикле for проверяются поочередно символы строки. Если символ – это английская буква, то она добавляется в стек. Иначе в переменную x записывается последний элемент стека, он же удаляется из стека, в переменную y записывается последний элемент стека и потом тоже удаляется из него. Далее в стек добавляется элемент, содержащий “(“ + y + оператор из строки + x + “)”. После выхода из цикла возвращается последний элемент стека.

**2.3 Декомпозиция:**

1. Проверка символа на принадлежность к английскому алфавиту

Предусловие. in\_x – символ из введенной строки

Постусловие. Возврат 1 или 0

bool Operator(char in\_x)

1. Возврат выражения в инфиксной форме

Предусловие. str – введенная строка

Постусловие. Возврат преобразованной строки

string in(string str)

**2.4 Код программы:**

#include <iostream>

#include <stack>

#include <string>

using namespace std;

bool Operator(char in\_x) {

return in\_x >= 'a' and in\_x <= 'z';

}

string in(string str) {

stack<string> stek;

for (int i = 0; str[i] != 0; i++) {

if (Operator(str[i])) {

string op(1, str[i]);

stek.push(op);

}

else {

string x = stek.top();

stek.pop();

string y = stek.top();

stek.pop();

stek.push("(" + y + str[i] + x + ")");

}

}

return stek.top();

}

int main() {

setlocale(LC\_ALL, "RUS");

string post = "xyz^+mnk/-\*p+";

cout << "Постфиксная нотация: " << post << endl;

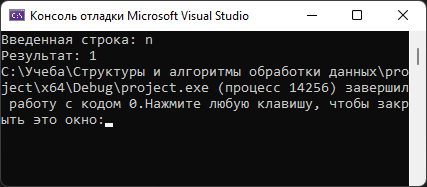
cout << "Инфиксная нотация: " << in(post);

return 0;

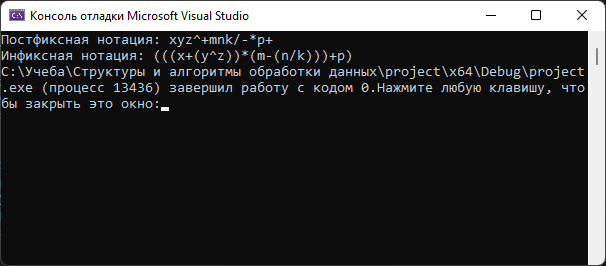
}

* 1. **Скриншоты:**

1. Тестирование функции декомпозиции: проверка символа



1. Тестирование функции декомпозиции: возврат выражения в инфиксной форме



ЗАДАНИЕ 3

Вычислить значение выражения, представленного в столбце 3, выполняя вычисление по образцу 4.

1. **Разработка программы задачи 3**
   1. **Постановка задачи**

Дано. Дана строка, содержащая постфиксную запись выражения

Результат. Вывод результата операций строки

* 1. **Описание модели решения**

Функции print(string post) передается параметр со значением исходной строки. Внутри этой функции поочередно подбираются элементы строки с помощью цикла for. Если текущий элемент – это цифра, то он добавляется в стек. Иначе переменным b и a присваиваются значения последнего и предпоследнего элемента стека соответственно, а затем удаляются из стека. В переменной res вызывается функция oper(a, elem, b), которой передаются параметры a, elem, b, где elem – текущий элемент(или знак оператора). В функции oper() выполняется операция над a и b. Далее res записывается в стек.

**3.3 Декомпозиция:**

1. Операция над двумя переменными

Предусловие. left, right – элементы из стека(цифры), operation – текущий элемент(оператор)

Постусловие. Возврат операции над a и b

int oper(string left, char operation, string right)

1. Вычисление значения выражения

Предусловие. post – исходная строка

Постусловие. Возврат значения выражения

int print(string post)

**3.4 Код программы:**

#include <iostream>

#include <stack>

#include <string>

using namespace std;

int oper(string left, char operation, string right) {

int a = stoi(left);

int b = stoi(right);

switch (operation) {

case '\*':

return a \* b;

case '/':

return a / b;

case '+':

return a + b;

case '-':

return a - b;

default:

return 0;

}

}

int print(string post) {

stack<string> stek;

for (char elem : post) {

if (isdigit(elem))

stek.push(string(1, elem));

else {

string b = stek.top();

stek.pop();

string a = stek.top();

stek.pop();

int res = oper(a, elem, b);

stek.push(to\_string(res));

}

}

return stoi(stek.top());

}

int main() {

setlocale(LC\_ALL, "ru");

string str = "723\*582/-+1-+";

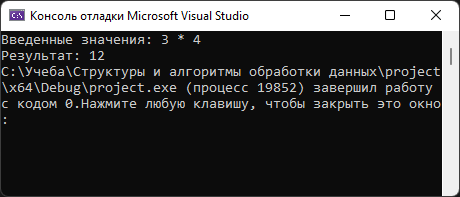
cout << "Исходно выражение: " << str << endl;

cout << "Результат: " << print(str) << endl;

}

* 1. **Скриншоты:**

1. Тестирование функции декомпозиции: операция над двумя переменными



1. Тестирование функции декомпозиции: вычисление значения выражения

