УO «Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

Лабораторная работа

**«Работа со стеком»**

Выполнила студентка

3 курса, 5 группы

Каспер Н.В.

Проверил

Бондарчик Е.Н.

Минск 2020

# **1. Постановка задачи**

Цель: Научиться использовать стек для управления разбором слов.

# **2. Краткие сведения**

**Стек –** абстрактный тип данных, который представляет собой список элементов, организованных по принципу LIFO

Файлы, которые используют стек – .exe

Файлы, которые не используют стек – .com (< 64кб).

Принцип стека – первым пришел, последним вышел (FILO).

2 вида передачи параметров в стек:

1. Стандартный (std) – верхушка стека – первый параметр и т.д.
2. Pascal-евый.

Разгрузка стека после завершения подпрограммы – важно (в ассемблере сами).

**Функции** стека:

\*загрузка параметров

\*сохранение состояния программы в момент прерывания

\*использование в компиляторах

# **3. Задание**

1. Написать программу для распознавания чисел с фиксированной точкой.

Программный код представлен в листинге 1.

|  |
| --- |
| bool double1(char \*word)  {  Stack<int> s(5);  s.free();  s.push(0);  int len = strlen(word);  for (int i = 0; i < len; i++)  {  int sost = s.pop();  if (sost == 0)  {  if (inttrue(word[i]) || word[i] == '-')  s.push(1);  else  {  s.push(-1);  }  }  else if (sost == 1)  {  if (inttrue(word[i]))  s.push(1);  else if (word[i] == '.')  {  s.push(2);  }  else  {  s.push(-1);  }  }  else if (sost == 2)  {  if (inttrue(word[i]))  s.push(3);  else  {  s.push(-1);  }  }  else if (sost == 3)  {  if (inttrue(word[i]))  s.push(3);  else if (word[i] == ' ')  {  cout << "Yes\n";  return true;  }  else  {  s.push(-1);  }  }  else if (sost == -1)  {  cout << "No\n";  return false;  }  else  {  s.push(-1);  }  }  if (s.pop() == -1)  {  cout << "No\n";  return false;  }  return true;  } |

1. Написать программу для распознавания чисел с плавающей точкой

Программный код представлен в листинге 2.

Листинг 2. Программный код для распознавания чисел с плавающей точкой

|  |
| --- |
| bool pleetouch2(char \*word) {  Stack<int> s(8);  s.free();  s.push(0);  int len = strlen(word);  for (int i = 0; i < len; i++)  {  int sost = s.pop();  if (sost == 0)  {  if (inttrue(word[i]) || word[i] == '-' || word[i] == '+')  s.push(1);  else  {  s.push(-1);  }  }  else if (sost == 1)  {  if (inttrue(word[i]))  s.push(1);  else if (word[i] == '.')  {  s.push(2);  }  else  {  s.push(-1);  }  }  else if (sost == 2)  {  if (inttrue(word[i]))  s.push(3);  else  {  s.push(-1);  }  }  else if (sost == 3)  {  if (inttrue(word[i]))  s.push(3);  else if (word[i] == 'e')  {  s.push(4);  }  else  {  s.push(-1);  }  }  else if (sost == 4)  {  if (word[i] == '+' || word[i] == '-')  s.push(5);  else  {  s.push(-1);  }  }  else if (sost == 5)  {  if (inttrue(word[i]))  s.push(6);  else  {  s.push(-1);  }  }  else if (sost == 6)  {  if (inttrue(word[i]))  s.push(6);  else if (word[i] == ' ')  {  cout << "Yes\n";  return true;  }  else  {  s.push(-1);  }  }  else if (sost == -1)  {  cout << "No\n";  return false;  }  else  {  s.push(-1);  }  }  if (s.pop() == -1)  {  cout << "No\n";  return false;  }  return true;  } |

1. Написать программу для распознавания комплексных чисел

Программный код представлен в листинге 3.

Листинг 3. Программный код для распознавания комплексных чисел

|  |
| --- |
| bool complix3(char \*word)  {  Stack<int> s(5);  s.free();  s.push(0);  int len = strlen(word);  for (int i = 0; i < len; i++)  {  int sost = s.pop();  if (sost == 0)  {  if (inttrue(word[i]) || word[i] == '-' || word[i] == '+')  s.push(1);  else  {  s.push(-1);  }  }  else if (sost == 1)  {  if (inttrue(word[i]))  s.push(1);  else if (word[i] == '-' || word[i] == '+')  {  s.push(2);  }  else  {  s.push(-1);  }  }  else if (sost == 2)  {  if (inttrue(word[i]))  s.push(2);  else if (word[i] = 'j')  {  s.push(3);  }  else  {  s.push(-1);  }  }  else if (sost == 3)  {  if (word[i] == ' ')  {  cout << "Yes\n";  return true;  }  else  {  s.push(-1);  }  }  else if (sost == -1)  {  cout << "No\n";  return false;  }  else  {  s.push(-1);  }  }  if (s.pop() == -1)  {  cout << "No\n";  return false;  }  return true;  } |

1. Написать программу для распознавания простейших алгебраических выражений типа a+b\*c (выражения не используют скобки, операндами являются одиночные буквы a и b)

Программный код представлен в листинге 4.

Листинг 4. Программный код для распознавания простейших алгебраических выражений

|  |
| --- |
| bool algebr4(char \*word)  {  Stack<int> s(3);  s.free();  s.push(0);  int len = strlen(word);  for (int i = 0; i < len; i++)  {  int sost = s.pop();  if (sost == 0)  {  if (word[i] == 'b' || word[i] == 'a')  s.push(1);  else  {  s.push(-1);  }  }  else if (sost == 1)  {  if (word[i] == '+' || word[i] == '-' || word[i] == '\*' || word[i] == ':')  s.push(0);  else if (word[i] == ' ')  {  cout << "Yes\n";  return true;  }  else  {  s.push(-1);  }  }  else if (sost == -1)  {  cout << "No\n";  return false;  }  else  {  s.push(-1);  }  }  if (s.pop() == -1)  {  cout << "No\n";  return false;  }  return true;  } |

1. Написать программу, которая определяет, что в выражении все скобки спарены

Программный код представлен в листинге 5.

Листинг 5. Программный код для распознавания правильности расставления скобок

|  |
| --- |
| bool is\_correct(const string& str)  {  stack< char > stack;  for (int i = 0; i < (int)str.length(); ++i)  {  if (str[i] == '(')  {  stack.push(str[i]);  }  else if (str[i] == ')')  {  if (stack.empty() || (stack.top() != '('))  {  return false;  }  stack.pop();  }  }  return stack.empty();  } |

Результат работы приложения:





