|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

по дисциплине: «Основы программирования»

|  |  |  |
| --- | --- | --- |
| Студент |  | Камалов Антон Павлович |
| Группа |  | РК6-15Б |
| Тип задания |  | Лабораторная работа №7 |
| Тема лабораторной работы |  | Функции ввода-вывода getchar() и putchar() |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \_**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2022 г.*

# Цель работы

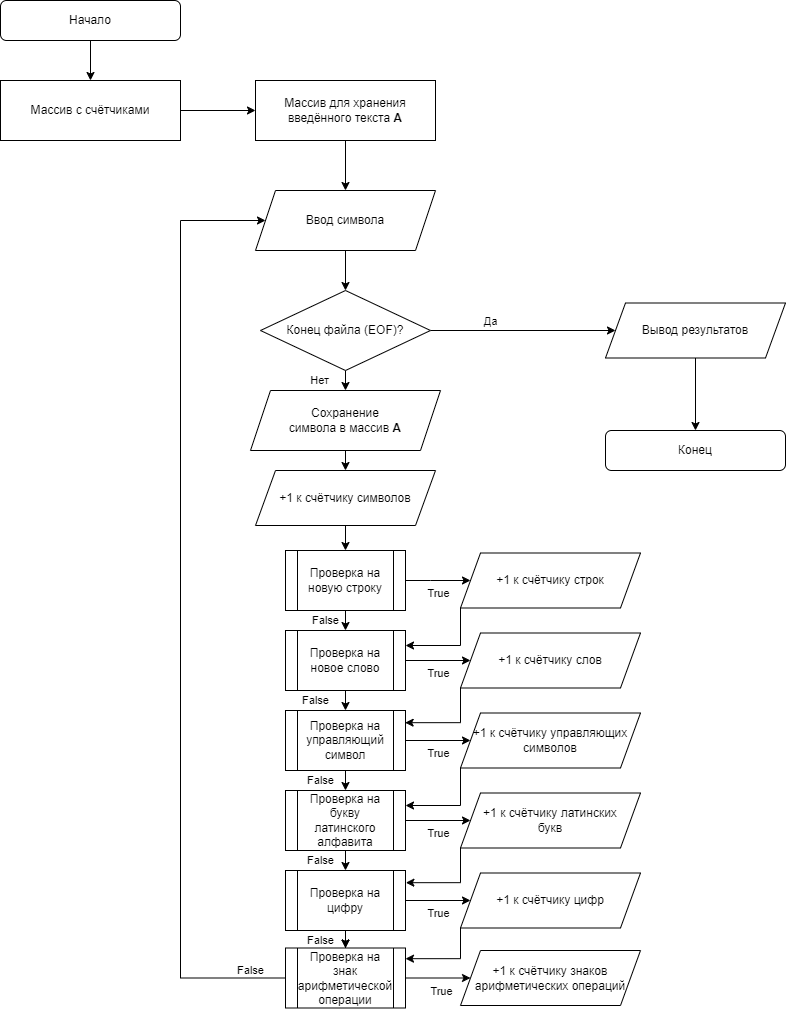
# Изучение буферизованного ввода-вывода и стандартных функций ввода-

# вывода getchar() и putchar().

# Ответы на контрольные вопросы

1. Буферизованный ввод-вывод (буферизация) - способ организации ввода-вывода в программе, позволяющий минимизировать число обращений к устройству.
2. Функция getchar() используется для ввода символа. Функция getchar() читает из стандартного устройства ввода один символ и возвращает его код в соответствии с таблицей кодирования. Если при попытке чтения обнаружен конец файла или ошибка, то getchar() возвращает значение EOF.
3. Функция putchar() используется для вывода символа. Функция putchar() выводит один символ в стандартный вывод , возвращает при успешном завершении код выводимого символа иначе EOF.
4. EOF - поименованная константа, определяющая признак конца файла.
5. В языке Си для хранения символьных строк используются символьные массивы.
6. “Нулевым” символом '\0'.

**Блок-схема программы**

****

# Текст программы

# #include <stdio.h>

# int count\_word(char c, char last){

# int x1 = (last == '\t') || (last == ' ') || (last == '\n');

# int x2 = (c != ' ') && (c != '\t') && (c != '\n');

# return x1 && x2;

# }

# int count\_cntr(char c){

# return (c >= 0 && c <= 31);

# }

# int count\_lat(char c){

# return (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z');

# }

# int count\_digit(char c){

# return (c >= '0' && c <= '9');

# }

# int count\_lines(char c){

# return (c == '\n');

# }

# int count\_op(char c){

# return (c == '+' || c == '-' || c == '%' || c == '/');

# }

# void out(char s[], int a[]){

# printf("\nResults:\n\n");

# printf("Source text:\n");

# for(int i = 0; i < a[0]; i++)

# putchar(s[i]); // посимвольный вывод текст

# printf("\nCharacters - %d\nLines - %d\nWords - %d\n", a[0],a[1],a[2]);

# printf("Control characters - %d\nLatin letters - %d\nDigits - %d\nOperations - %d\n", a[3],a[4],a[5],a[6]);

# }

# int main()

# {

# int c;

# int a[7] = {0, 0, 0, 0, 0, 0, 0}; // 0 - символы, 1 - строки, 2 - слова, 3 - упр. символы, 4 - лат. алфавит, 5 - цифры, 6 - знаки операций

# char s[1024], last = ' ';

# printf("Enter some text:\n");

# while ((c=getchar())!=EOF){

# s[a[0]] = c;

# a[0]+=1;

# a[1]+=count\_lines(c);

# a[2]+=count\_word(c, last);

# a[3]+=count\_cntr(c);

# a[4]+=count\_lat(c);

# a[5]+=count\_digit(c);

# a[6]+=count\_op(c);

# last = c;

# }

# out(s, a);

# }

# Тестовые примеры

|  |
| --- |
| blah blah blah  do you understaand?  1000-7  kira  Ignorance is strength |
| 500-86=414  wait  please wait  potom  Freedom is slavery |
| я узкий  good  i want to drink |
| 999/3=333  fact  \*\*\*@@@!@))++ |
| last test  ehhhh  Ok  223724242  sdf  GAME IS GAME! |

# Результаты отладки

После первой компиляции была найдена синтаксическая ошибка: в вызове функции count\_lines() отсутствовал аргумент.

# Результаты работы программы

# Выводы по работе

Я изучил буферизованный ввод-вывод и стандартные функции ввода-

вывода getchar() и putchar().