



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК "Информатика и управление"

КАФЕДРА ИУК5 "Системы обработки информации"

ЛАБОРАТОРНАЯ РАБОТА №3

«ДЕЙСТВИЯ НАД МАССИВАМИ И СТРОКАМИ»

ДИСЦИПЛИНА: «Основы программирования»

Выполнил: студент гр.

_____(Подпись) _____(Ф.И.О.)

_____(Подпись) _____(Ф.И.О.)
(Кондратьева С. Д.)

Проверила:

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 202_

Цель: формирование практических навыков разработки эффективных алгоритмов обработки массивов и строк

Задачи:

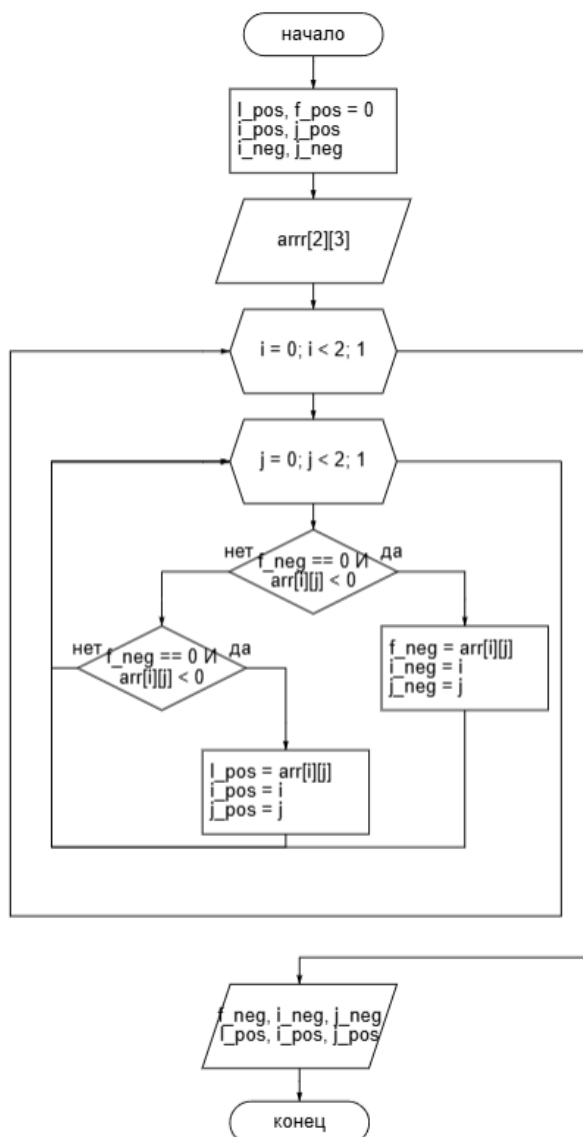
1. Овладеть навыками построения эффективных алгоритмов обработки массивов и строк.
2. Овладеть навыками выбора оптимальных алгоритмов программирования.
3. Овладеть навыками описания основных этапов построения алгоритмов

Порядок выполнения

Вариант 1

Задание 1:

Дана матрица $A(n \times m)$. Найти порядковые номера первого отрицательного и последнего положительного элемента (если таковые имеются). Значение элементов и их порядковые номера вывести на экран или выдать соответствующее сообщение.



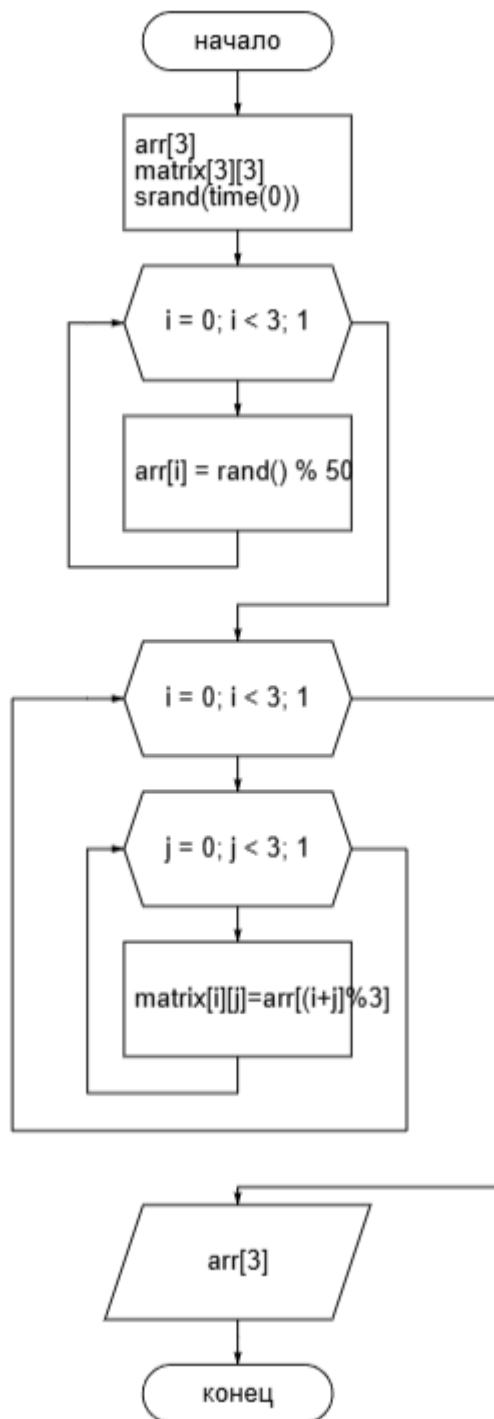
```
#include <iostream>
2
3 int main(void)
4 {
5     int arr[2][3] = {};
6     int l_pos, f_neg = 0;
7     int i_pos, j_pos, i_neg, j_neg;
8
9     for (int i = 0; i < 2; i++)
10         for (int j = 0; j < 3; j++)
11             std::cin >> arr[i][j];
12
13     for (int i = 0; i < 2; i++)
14         for (int j = 0; j < 3; j++)
15         {
16             if (f_neg == 0 && arr[i][j] < 0)
17             {
18                 f_neg = arr[i][j];
19                 i_neg = i;
20                 j_neg = j;
21             }
22             if (arr[i][j] > 0)
23             {
24                 l_pos = arr[i][j];
25                 i_pos = i;
26                 j_pos = j;
27             }
28         }
29
30     std::cout << "First_negative: " << f_neg << " | i: " << i_neg << " j: " << j_neg << std::endl;
31     std::cout << "Last positive: " << l_pos << " | i: " << i_pos << " j: " << j_pos << std::endl;
32     return 0;
33 }
```

1
2
3
-4
-5
6
First_negative: -4 | i: 1 j: 0
Last positive: 6 | i: 1 j: 2

Задание 2:

Дан массив $A(n)$. Построить матрицу $A(n \times n)$ вида

$$\begin{bmatrix} a_1 & a_2 & \dots & a_{n-1} & a_n \\ a_2 & a_3 & \dots & a_n & a_1 \\ \dots & \dots & \dots & \dots & \dots \\ a_n & a_1 & \dots & a_{n-2} & a_{n-1} \end{bmatrix}.$$

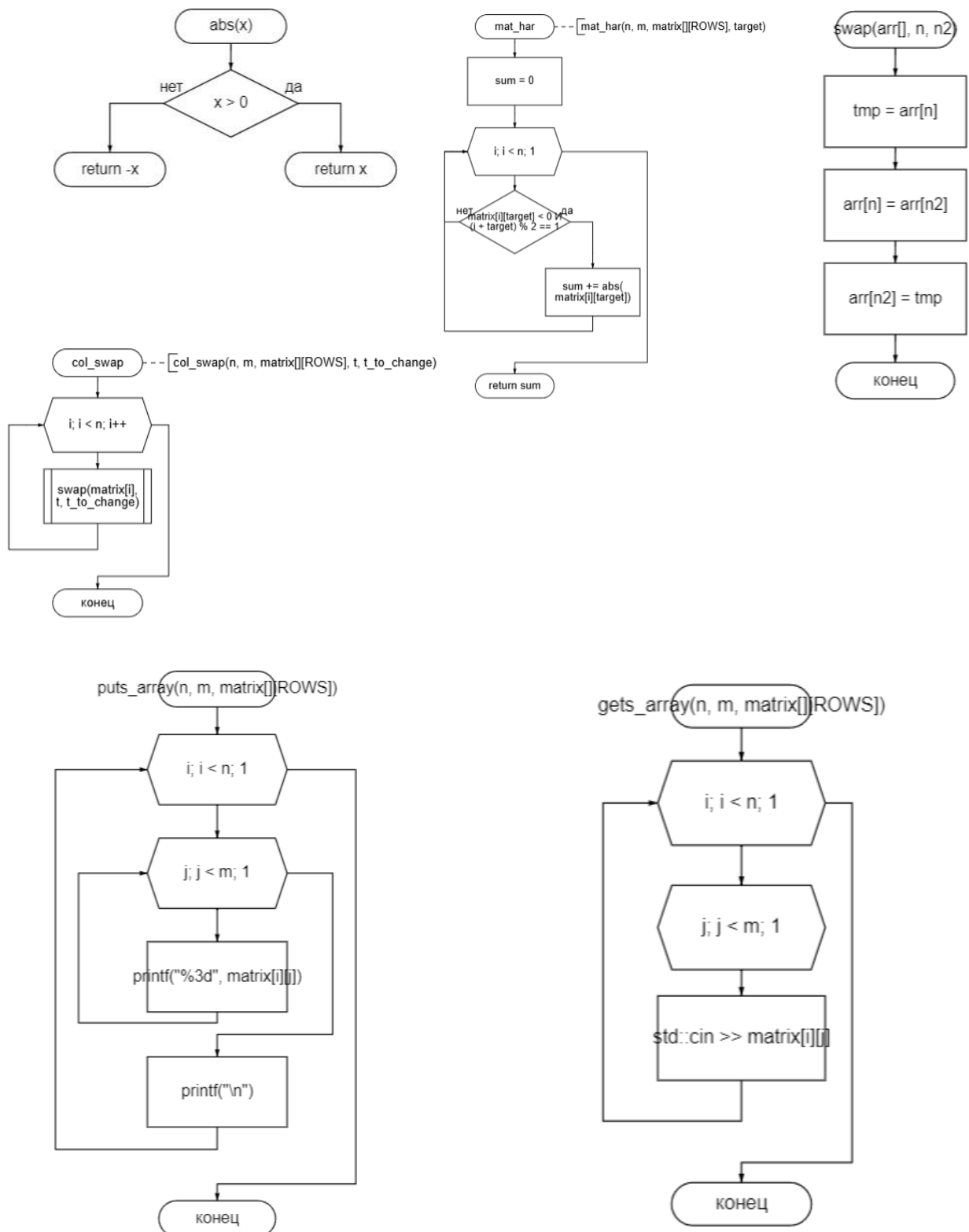


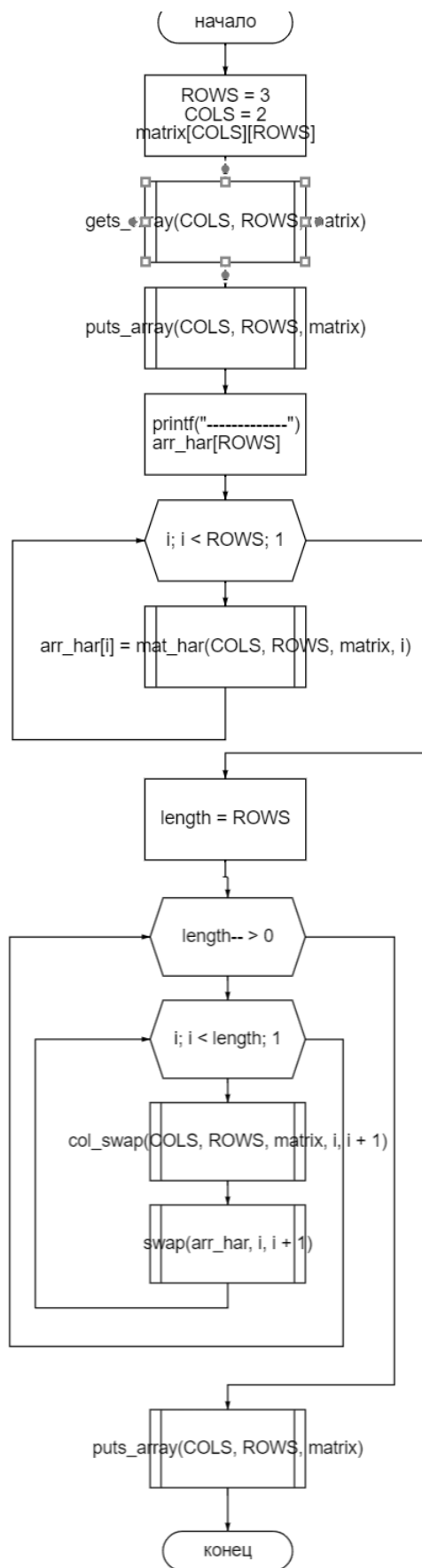
```
1  #include <iostream>
2  #include <random>
3
4  int main(void)
5  {
6      int arr[3], matrix[3][3];
7
8      srand(time(0));
9      for (int i = 0; i < 3; i++)
10         arr[i] = rand() % 50;
11
12     for (int i = 0; i < 3; i++)
13         for (int j = 0; j < 3; j++)
14             matrix[i][j] = arr[(i + j) % 3];
15
16     for (int i = 0; i < 3; i++)
17     {
18         for (int j = 0; j < 3; j++)
19             std::cout << matrix[i][j] << ' ';
20         std::cout << std::endl;
21     }
22     return 0;
23 }
```

```
0 31 19
31 19 0
19 0 31
```

Задание 3:

Характеристикой столбца целочисленной матрицы $A(n \times m)$ назовем сумму модулей его отрицательных нечетных элементов. Переставляя столбцы заданной матрицы, расположить их в соответствии с ростом их характеристик.





```

1  #include <iostream>
2  #include <stdio.h>
3  #define ROWS 3
4  #define COLS 2
5
6  int abs(int x)
7  {
8      return (x >= 0) ? x : -x;
9  }
10
11 int mat_har(int n, int m, int matrix[][ROWS], int target) //Характеристика числа по заданию
12 {
13     int sum = 0;
14     for (int i = 0; i < n; i++)
15         if (matrix[i][target] < 0 && (i + target) % 2 == 1)
16             sum += abs(matrix[i][target]);
17     return sum;
18 }
19
20 void swap(int arr[], int n, int n2) //Меняем местами значения 2 переменных в одном массиве
21 {
22     int tmp = arr[n];
23     arr[n] = arr[n2];
24     arr[n2] = tmp;
25 }
26
27 void col_swap(int n, int m, int matrix[][ROWS], int t, int t_to_change) //Меняем столбцы в матрице
28 {
29     for (int i = 0; i < n; i++)
30     {
31         swap(matrix[i], t, t_to_change);
32     }
33 }
34
35 void puts_array(int, int, int matrix[][ROWS]); //Для вывода 2 мерного массива
36
37 void puts_array(int, int, int matrix[][ROWS]); //Для вывода 2 мерного массива
38
39 void gets_array(int, int, int matrix[][ROWS]); //Для ввода 2 мерного массива
40
41 int main(void)
42 {
43     int matrix[COLS][ROWS];
44     gets_array(COLS, ROWS, matrix); //Вводим 0 -2 0 -3 0 -1
45     puts_array(COLS, ROWS, matrix);
46     printf("-----\n");
47
48     int arr_har[ROWS]; //Сохраняем характеристики столбцов (при перемещении столбцов они могут перестать подходить по условию)
49     for (int i = 0; i < ROWS; i++)
50     {
51         arr_har[i] = mat_har(COLS, ROWS, matrix, i);
52     }
53
54     int length = ROWS;
55     while (length > 1) //Пузырьковая сортировка по возрастанию
56     {
57         for (int i = 0; i < length; i++)
58             if (arr_har[i] > arr_har[i + 1]) //Если характеристика в arr_har больше следующей то меняем местами столбцы и значения характеристики
59             {
60                 col_swap(COLS, ROWS, matrix, i, i + 1);
61                 swap(arr_har, i, i + 1);
62             }
63         length--;
64     }
65
66     puts_array(COLS, ROWS, matrix);
67
68     return 0;
69 }
70
71 void puts_array(int n, int m, int matrix[][ROWS]) //Определение для функции вывода
72 {
73     for (int i = 0; i < n; i++)
74     {
75         for (int j = 0; j < m; j++)
76             printf("%3d", matrix[i][j]);
77         printf("\n");
78     }
79 }
80
81 void gets_array(int n, int m, int matrix[][ROWS]) //Определение для функции ввода
82 {
83     for (int i = 0; i < n; i++)
84         for (int j = 0; j < m; j++)
85             std::cin >> matrix[i][j];
86 }

```

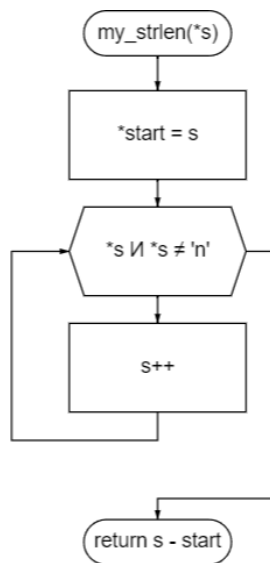
```

0
-2
0
-3
0
-1
0 -2 0
-3 0 -1
-----
0 -2 0
-1 0 -3

```

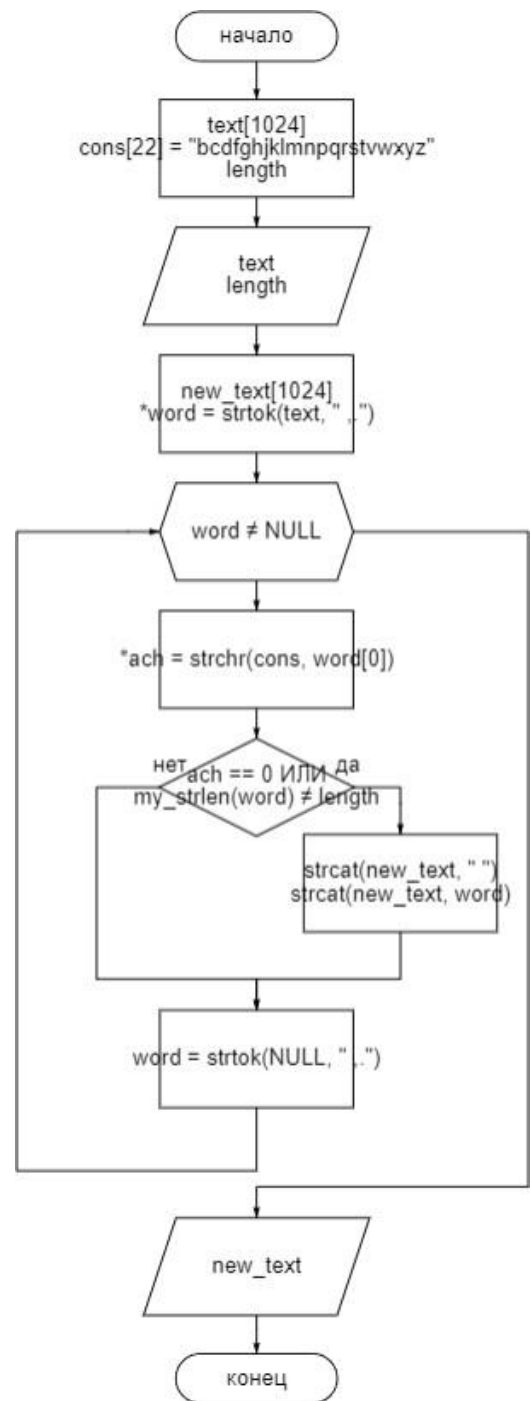
Задание 4:

Из текста удалить все слова заданной длины, начинающиеся с согласных букв.



```

1  #include <iostream>
2  #include <stdio.h>
3  #include <string.h>
4
5  size_t my_strlen(const char *s) //Для подсчета длины без учета переноса строки
6  {
7      const char *start = s;
8      while (*s && *s != '\n')
9          s++;
10     return s - start;
11 }
12
13 int main(void)
14 {
15
16     char text[1024];
17     const char cons[22] = "bcdfghjklmnpqrstvwxyz"; //строка согласных
18     int length;
19
20     std::cout << "Enter text: ";
21     fgets(text, 1024, stdin); //Считываем строку из входного потока
22
23     std::cout << "Select length: ";
24     std::cin >> length;
25
26
27
28     char new_text[1024];
29     char *word = strtok(text, " ,."); //Получаем указатель на первое слово
30     while (word != NULL)
31     {
32         char *ach = strchr(cons, word[0]); //Ищем 1 букву в строке согласных
33         if (ach == 0 || my_strlen(word) != length)
34         {
35             strcat(new_text, " "); //добавляем пробел между словами
36             strcat(new_text, word); //добавляем слово в наш текст
37         }
38         word = strtok(NULL, " ,."); //Переходим на следующее слово
39     }
40
41     puts(new_text);
42
43     return 0;
44 }
  
```

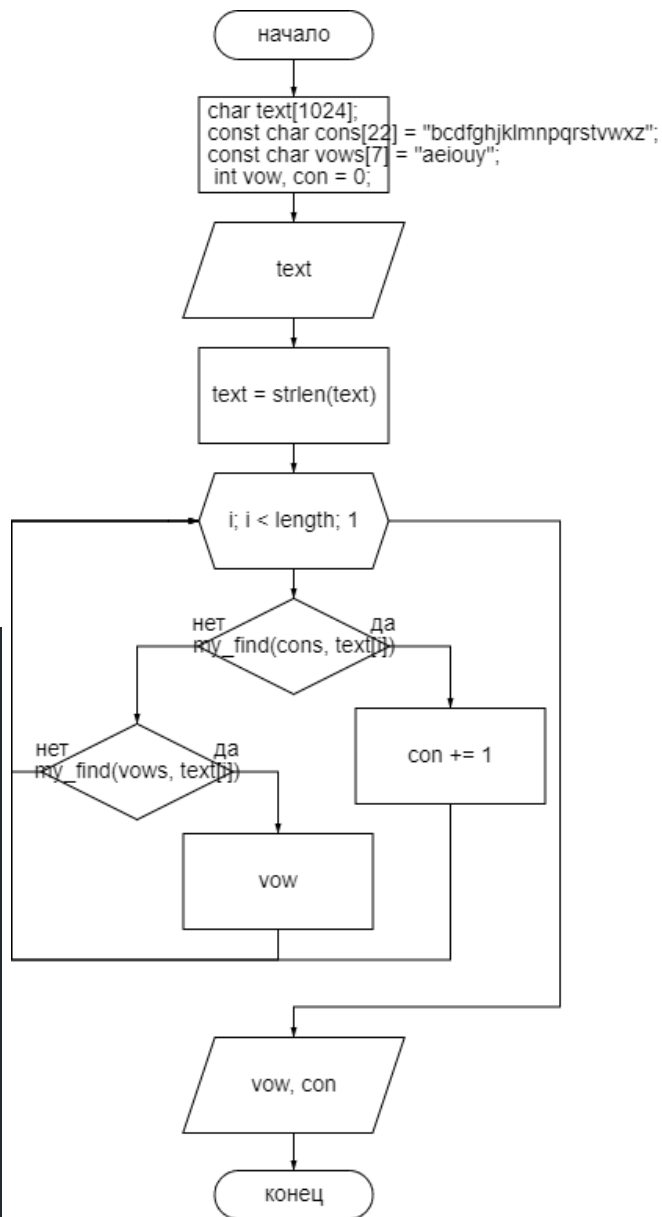
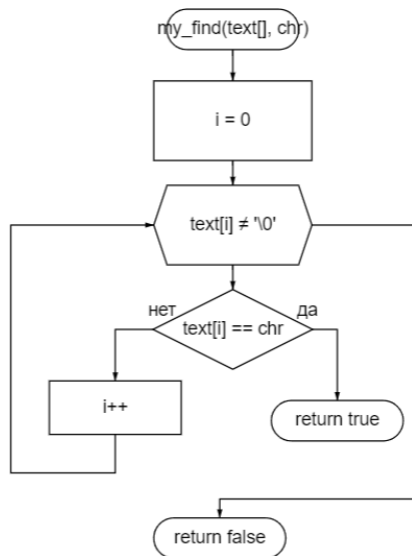


```

Enter text: hhhh alkf hehe
Select length: 4
hhhh alkf
  
```


Задание 5:

Найти каких букв в тексте больше - гласных или согласных.



```

1  #include <iostream>
2  #include <string.h>
3
4  bool my_find(const char text[], char chr) //ищем букву в переданной строке
5  {
6      int i = 0;
7      while (text[i] != '\0')
8      {
9          if (text[i] == chr)
10             return true;
11         i++;
12     }
13     return false;
14 }
15
16 int main(void)
17 {
18     char text[1024];
19     const char cons[22] = "bcdfghjklmnpqrstvwxyz";
20     const char vows[7] = "aeiouy";
21     int vow, con = 0;
22     fgets(text, 1024, stdin);
23
24     int length = strlen(text);
25     for (int i = 0; i < length; i++)
26     {
27         if (my_find(cons, text[i]))
28             con += 1;
29         if (my_find(vows, text[i]))
30             vow += 1;
31     }
32
33     std::cout << "vow: " << vow << std::endl;
34     std::cout << "con: " << con << std::endl;
35     return 0;
36 }
37
  
```

```

aeebbdds
vow: 3
con: 6
  
```

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ:

1. Дайте определение понятию «массив»

Массив – это совокупность определенного количества однотипных переменных, имеющих одно имя

2. Укажите, как называют переменные массива

Переменные массива называют *элементами массива*.

3. Укажите, с помощью чего можно получить доступ к элементу массива

Каждый элемент имеет свой уникальный **индекс** – свой порядковый номер. Используя индекс, мы можем обращаться к конкретному элементу.

4. Приведите примеры объявления массивов

```
int a[5] = {1, 2, 3, 4, 5};
```

5. Раскройте значение термина «строка»

Строки - это такие же массивы, как и рассмотренные выше, но хранят они не числовые данные, а символьные.

6. Укажите различия массивов и строк

Строки хранят только символы и последний элемент всегда '\0'. У массивов же ограничений нет.

7. Приведите примеры объявления строк

```
char st[] = "hello!";
```

8. Перечислите и опишите основные операции работы со строками

Основные операции со строками: конкатенация, объединение, разделение, поиск, удаление и замена символов и подстрок.

9. Укажите, что необходимо сделать, чтобы присвоить элементу массива значение '\0'

Найти последний индекс в массиве и присвоить элементу '\0'

ВЫВОД: Были сформированы практические навыки разработки эффективных алгоритмов обработки массивов и строк.