



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) (Кондратьева С. Д.)  
(Ф.И.О.)

Проверила:

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 202

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

**Задачи:**

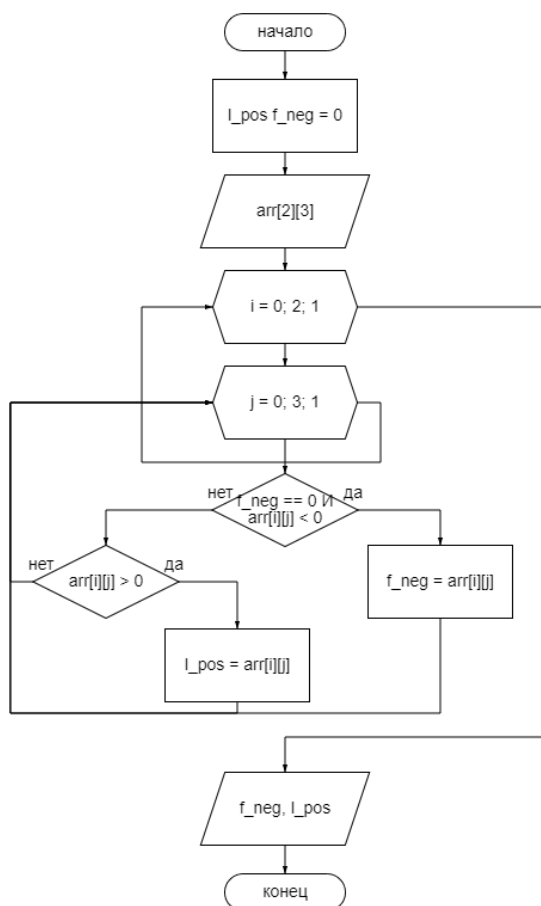
1. Овладеть навыками построения эффективных алгоритмов обработки массивов и строк.
2. Овладеть навыками выбора оптимальных алгоритмов программирования.
3. Овладеть навыками описания основных этапов построения алгоритмов

# 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.



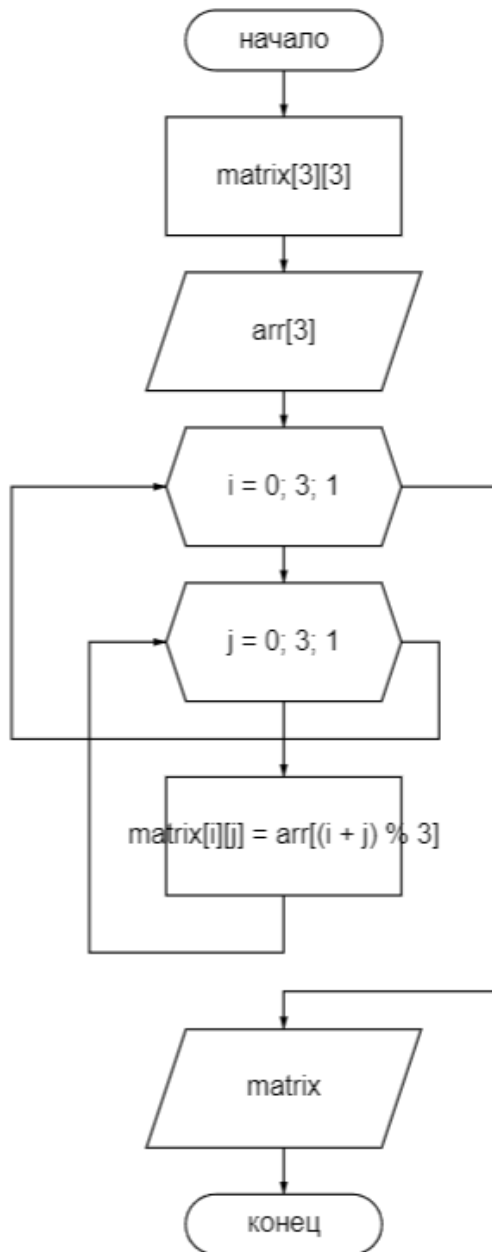
```
1  #include <iostream>
2
3  int main(void)
4  {
5      int arr[2][3] = {};
6      int l_pos, f_neg = 0;
7
8      for (int i = 0; i < 2; i++)
9          for (int j = 0; j < 3; j++)
10             std::cin >> arr[i][j];
11
12     for (int i = 0; i < 2; i++)
13         for (int j = 0; j < 3; j++)
14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

```
1
4
-10
4
5
6
First_negative: -10
Last positive: 6
```

## Задание 2:

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

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

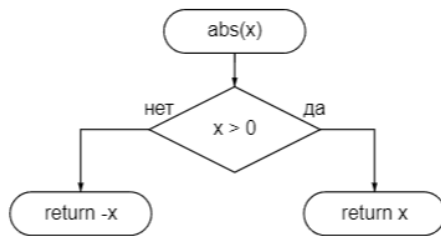


```
1  #include <iostream>
2
3  int main(void)
4  {
5      int arr[3], matrix[3][3];
6
7      for (int i = 0; i < 3; i++)
8          std::cin >> arr[i];
9
10     for (int i = 0; i < 3; i++)
11         for (int j = 0; j < 3; j++)
12             matrix[i][j] = arr[(i + j) % 3];
13
14     for (int i = 0;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

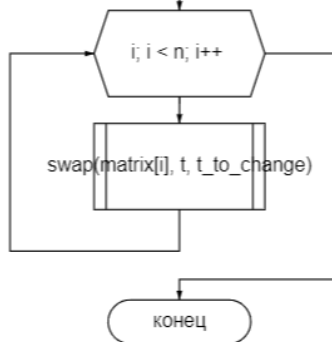
```
1
2
3
1 2 3
2 3 1
3 1 2
```

### Задание 3:

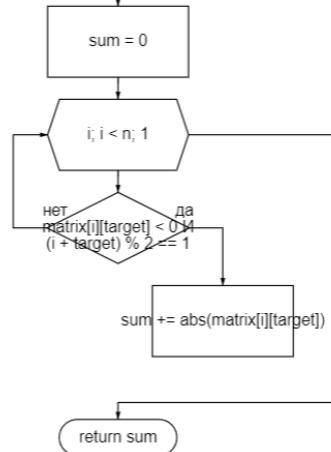
Характеристикой столбца целочисленной матрицы  $A(n \times m)$  назовем сумму модулей его отрицательных нечетных элементов. Переставляя столбцы заданной матрицы, расположить их в соответствии с ростом их характеристик.



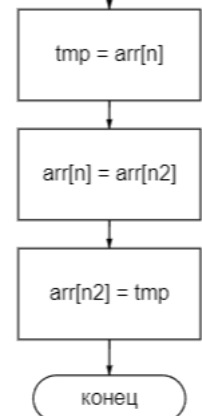
col\_swap(n, m, matrix[][ROWS], t, t\_to\_change)



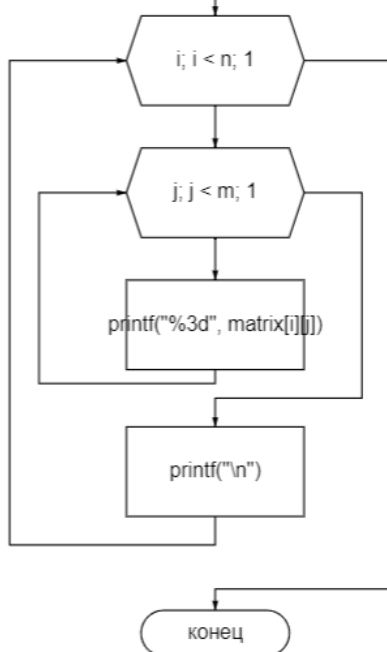
mat\_har(n, m, matrix[][ROWS], target)



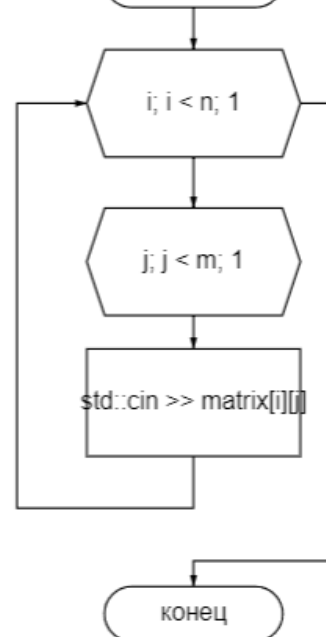
swap(arr[], n, n2)

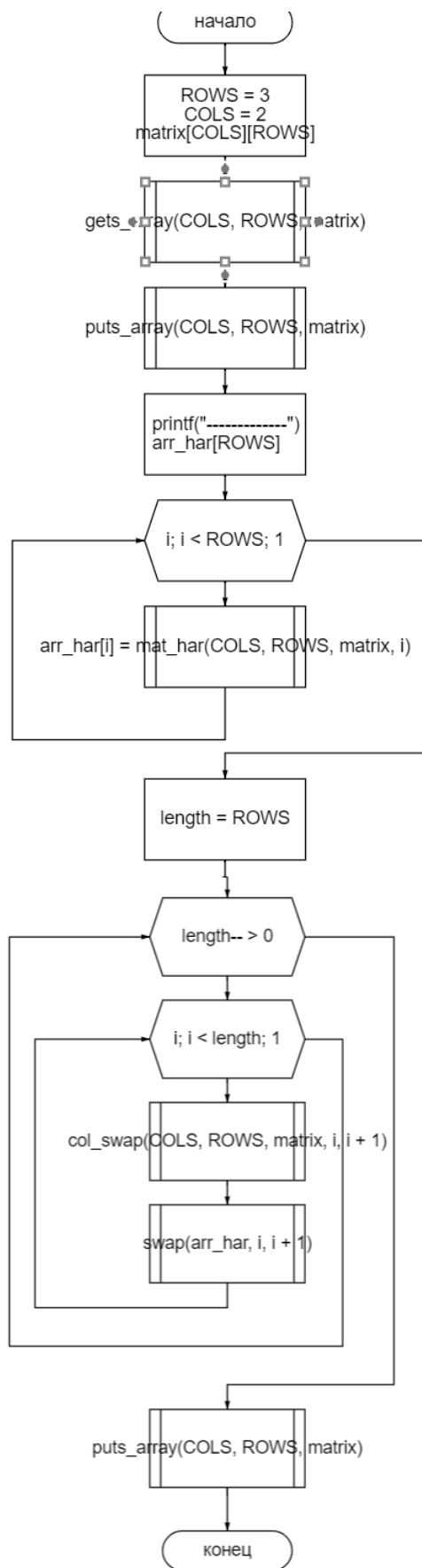


puts\_array(n, m, matrix[][ROWS])



gets\_array(n, m, matrix[][ROWS])





```

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
45     gets_array(COLS, ROWS, matrix); //Вводим 0 -2 0 -3 0 -1
46     puts_array(COLS, ROWS, matrix);
47     printf("-----\n");
48
49     int arr_har[ROWS]; //Сохраняем характеристики столбцов (при перемещении столбцов они могут перестать подходить по условию)
50     for (int i = 0; i < ROWS; i++)
51     {
52         arr_har[i] = mat_har(COLS, ROWS, matrix, i);
53     }
54
55     int length = ROWS;
56     while (length--) //Пузырьковая сортировка по возрастанию
57     {
58         for (int i = 0; i < length; i++)
59             if (arr_har[i] > arr_har[i + 1]) //Если характеристика в arr_har больше следующей то меняем местами столбцы и значения характеристик
60             {
61                 col_swap(COLS, ROWS, matrix, i, i + 1);
62                 swap(arr_har, i, i + 1);
63             }
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 }

```

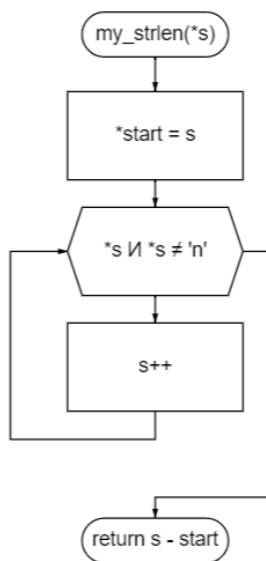
```

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

```

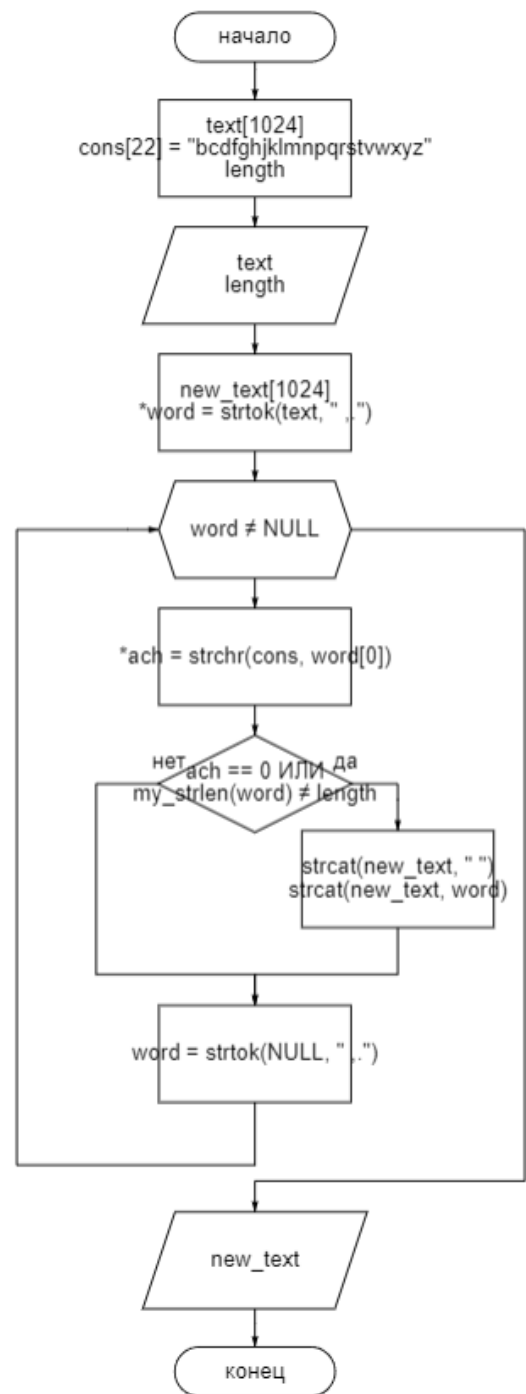
## Задание 4:

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             strcat(new_text, " "); //добавляем пробел между словами
35             strcat(new_text, word); //добавляем слово в наш текст
36         }
37         word = strtok(NULL, " ,."); //Переходим на следующее слово
38     }
39
40     puts(new_text);
41
42     return 0;
43 }
  
```



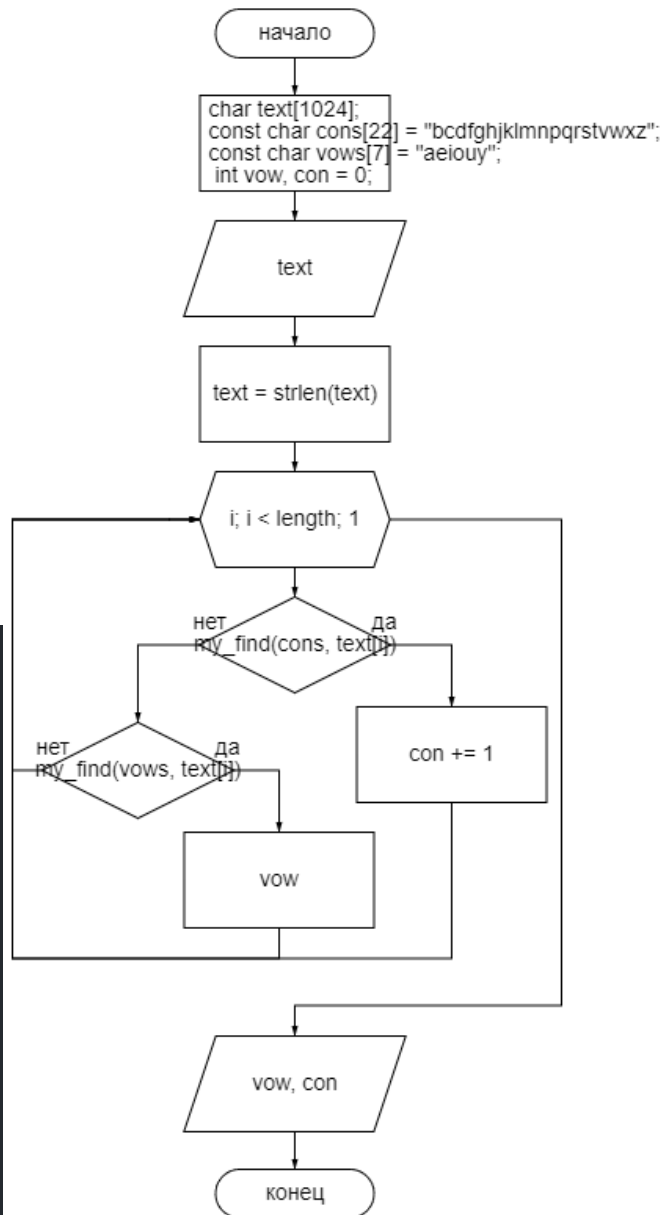
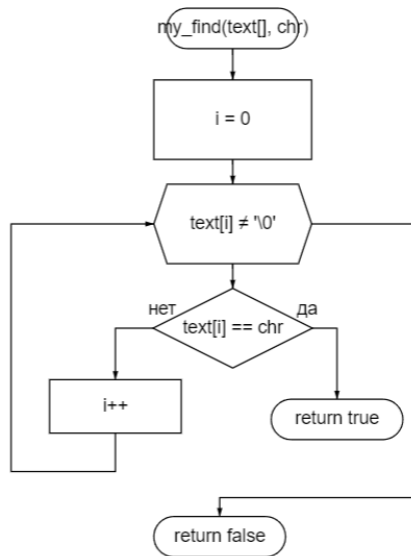
```

Enter text: hhhhh alkf hehe
Select length: 4
hhhhh 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] = "bcd fghjklmnpqrstvwxyz";
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 }
  
```

```

aeebbddss
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.