**ва**

Перечень экзаменационных вопросов

***Определения***

# **Алгоритм** - точное предписание исполнителю совершить определенную последовательность действий для достижения поставленной цели за определенное количество шагов.

# **Свойства алгоритма**

**Дискретность** - Следующий шаг алгоритма выполняется после того, как выполнен предыдущий шаг.

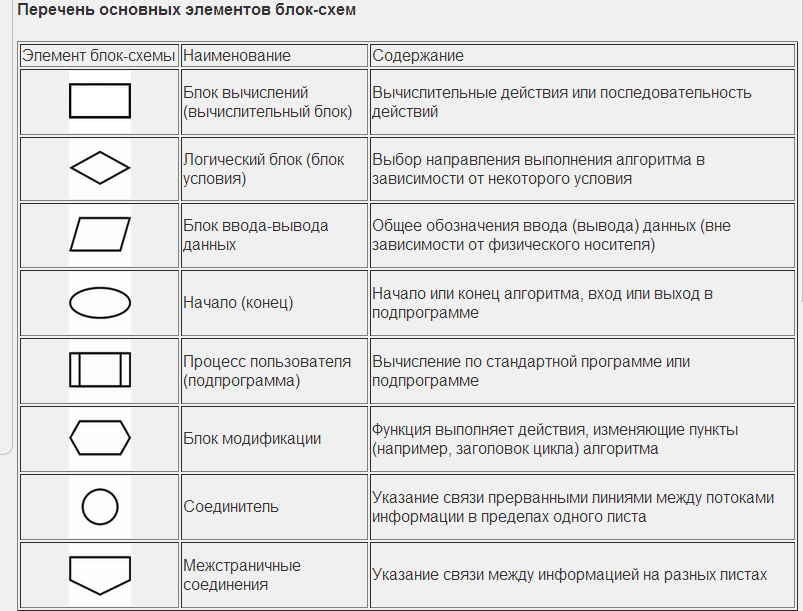
**Понятность**- Каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола.

**Результативность** - Алгоритм должен приводить к решению задачи за конечное число шагов.

**Массовость** - должен быть применимым для некоторого класса задач, различными только исходными данными.

**Детерминированность** - каждая команда алгоритма определяет однозначность действий.

# Блоки алгоритмов

****

# Виды алгоритмов :

## **Линейный алгоритм** - набор команд (указаний) выполняемых во времени последовательно друг за другом.

## **Разветвляющийся алгоритм** - алгоритм, содержащий хотя бы одно условие, в результате проверки которого эвм обеспечивает переход на один из двух возможных шагов.

**Виды:**

1. If/else
2. switch(case)
3. GoTo (перейти туда )

## **Цикл** - это алгоритм, в котором некоторая последовательность шагов выполняется несколько раз.

**3 вида циклов**

1) Предусловие **While**

2) Постусловие  **Repeat**

3) Фиксированное количество повторов  **For**

Нормальный ход работы цикла может быть нарушен процедурами **Break** и **Continue**.

# **Данные** - это любая информация, представленная в формализованном виде и пригодна для обработки алгоритма

Есть 2 вида данных

1. **Искомые -** известные перед началом алготорма( Pi = 3.14)
2. **Конечные (выходные) -** данные являющиеся результатом действия алгоритма.

Так же данные могут делаться на

1. **Переменная -** данными можно крутить как угодно
2. **Константа** - значение которое устанавливается и не может меняется в процессе выполнения
3. **Идентификатор -** Последовательность букв и цифр (начало с буквы)

**Листинг**-Процедура вычисления суммы элементов верхней треугольной матрицы

**Теоретические вопросы**

# **1. Дайте определение и объясните принцип функционирования оператора выбора (пример).**

**Понятие:** если условие верно (true) то выполняется одно действие , а если не выполняется (false)то выполняется другое действие

**If/else switch(case) GoTo(**также оператор безусловного перехода**)** (переходит к определенной строчке (может выполняться бесконечно))

**Пример:**

Если пар больше чем 1 то надо идти, если нет то можно поспать

# 2. Дайте определение и объясните принцип функционирования оператора цикла типа пока (пример).

Цикл пока (**While**)- Предусловие- пока проверяемое условие истинно мы делаем какое то действие.

Пока макароны белые Абдул их жарит. (условие = макароны белые?)

# **3. Дайте определение и объясните принцип функционирования оператора цикла типа для (пример).**

Цикл для -(**For**) для того чтобы F стала = 15

Фиксированное количество повторов **For-** Цикл выполняется столько сколько ему задали

# 4. Дайте определение и объясните принцип функционирования оператора присваивания, инкремента, декремента.

В Lazarus оператор присваивания выглядит так ( := ) , знак присваивания ПРИСВАИВАЕТ значение в переменную )))

Инкремент- увеличение переменной (на +1)

Декремент – уменьшение переменной (на -1)

## **5. Дайте определение и объясните принцип функционирования операторов переходов.**

Понятие: если условие верно (true) то выполняется одно действие , а если не выполняется (false)то выполняется другое действие

**If/else switch(case) GoTo(**также оператор безусловного перехода**)** (переходит к определенной строке (может выполняться бесконечно))

## **6. Дайте определение и объясните принцип функционирования условного оператора (пример).**

Понятие: если условие верно (true) то выполняется одно действие , а если не выполняется (false)то выполняется другое действие

**If/else switch(case)**

**Пример:**

Компьютер включен тогда можно играем, если нет то можно посмотреть ютуб. (Условие = пк включен?)

# **7. Дайте определение и приведите примеры идентификаторов.**

**Идентификатор -** Последовательность букв и цифр (начало с буквы). Идентификаторами задают имена переменных и константа

Пример:

* Номер паспорта (EX 1234 5678910)
* Номер машины (а123бв 456)

# **8. Дайте определение многомерным массивам и объясните принципы работы с ними.**

Многомерный массив — это массив, элементы которого являются массивы

Двумерный массив



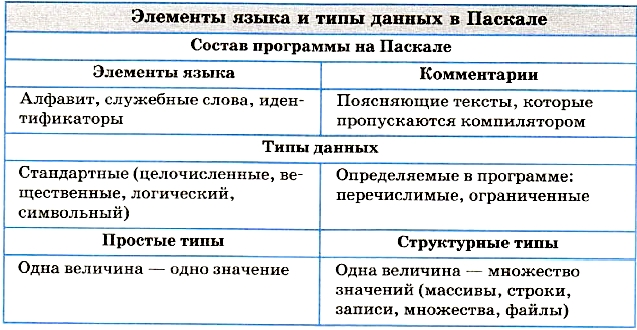
книга(массив) -> в ней много листов(массив в массиве) -> на листе много строк(массив в массиве в массиве)

# **9. Дайте определение одномерным массивам и объясните принципы работы с ними.**

**Одномерный массив** — это фиксированное количество элементов одного и того же типа, объединенных одним именем, где каждый элемент имеет свой номер.(index)

**Принцип работы:** Мы обращаемся к массиву и запрашиваем у него индекс. Под этим индексом стоит какое то число, вот его значение мы и берем.

# **10. Дайте определение основным элементам языка, основным типам данных.**

****

# **11. Дайте определение понятия алгоритма, охарактеризуйте свойства алгоритмов**.

Точное предписание исполнителю совершить определенную последовательность действий для достижения поставленной цели за определенное количество шагов.

1. **Дискретность** (прерывность, раздельность) - Следующий шаг алгоритма выполняется после того, как выполнен предыдущий шаг.

Например, подъем по этажам. Не поднявшись на второй, не поднимешься на третий.

1. **Понятность**(определенность) - Каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Например, инструкция на русском.
2. **Результативность**(конечность) - Алгоритм должен приводить к решению задачи за конечное число шагов. Например, алгоритм деления дробей.
3. **Массовость** - должен быть применимым для некоторого класса задач, различными только исходными данными. Например, алгоритм арифметических действий.
4. **Детерминированность**(однозначность) - каждая команда алгоритма определяет однозначность действий. Например, посадка и взлёт самолёта или покупка продуктов строго по списку.

# **12. Дайте определение языка программирования, элементов языка программирования.**

Язы́к программи́рования — [формальный язык](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA), предназначенный для записи [компьютерных программ](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)

1. ***лексика*** *-совокупность правил образования цепочек символов (лексем), образующих идентификаторы (переменные и метки), операторы, операции и другие лексические компоненты языка.*
2. ***алфавит*** *-совокупность символов, отображаемых на устройствах печати и экранах и/или вводимых с клавиатуры терминала*

# **13. Дайте определение языка программирования. Охарактеризуйте поколения языков программирования.**

Язы́к программи́рования — [формальный язык](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA), предназначенный для записи [компьютерных программ](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)

Существует несколько поколений:

1. Машинные
2. Ассемблеры
3. Высокоуровневые языки
4. Непроцедурные, Объектно-ориентированные, языки запросов, параллельные
5. Языки искусственного интеллекта, экспертных систем и баз знаний, естественные языки

# **14. Дайте сравнительную характеристику операторов цикла.**

Цикл с предусловием **while** - Цикл работает пока условие true

Цикл с постусловием **repeat**- Цикл работает пока условие false

Цикл со счетчиком **for**- Цикл выполняется столько сколько ему задали

Нормальный ход работы цикла может быть нарушен процедурами **Break** и **Continue**.

# **15. Дайте формулировку понятия данные. Расскажите о видах данных.**

# 

**Данны**е - это любая информация, представленная в формализованном виде и пригодна для обработки алгоритма

Есть 2 вида данных

1. **Искомые -** известные перед началом алготорма( Pi = 3.14)
2. **Конечные (выходные) -** данные являющиеся результатом действия алгоритма.

Так же данные могут делаться на

1. **Переменная -** данными можно крутить как угодно
2. **Константа** - значение которое устанавливается и не может меняется в процессе выполнения
3. **Идентификатор -** Последовательность букв и цифр (начало с буквы)

# **16. Дайте характеристику критериям качества алгоритма.**

**Качественный алгоритм имеет свойства**

1. Дискретность
2. Понятность
3. Результативность
4. Массовость
5. Детерминированность
6. Логичность

# **17. Назовите и дайте характеристику направлениям программирования.**

Основные направления программирования:

**Техническое программирование** – изучает принципы и способы построения программ, имея конечной целью автоматический (с помощью ЭВМ) синтез программ для решения любых задач;

**Системное программирование** – направлено на создание комплексов программ длительного и массового использования, обеспечивающих пользователям максимум удобств и возможностей (с помощью посредников или самостоятельно) для применения вычислительной техники при решении основного множества универсальных задач;

**Прикладное программирование** – обслуживает конкретные (в т.ч. уникальные или узкоспециальные) потребности в применении ЭВМ представителями различных областей науки и практики.

## **18. Объясните методы структурного программирования.**

1. алгоритм и программа должны составляться поэтапно (по шагам).

2. сложная задача должна разбиваться на достаточно простые части, каждая из которых имеет один вход и один выход.

3. логика алгоритма и программы должна опираться на минимальное число достаточно простых базовых управляющих структур*.*

Алгоритм пишется сверху-вниз

**Алгоритмы имеют 3 структурных вида**

1. Линейный
2. Разветвляющийся
3. Циклический

# 19. Объясните основные понятия алгоритмизации (алгоритм, алгоритмизация, исполнитель алгоритма).

**Алгоритм** - точное предписание исполнителю совершить определенную последовательность действий для достижения поставленной цели за определенное количество шагов.

**Исполнитель алгоритма** — это некоторая абстрактная или реальная система, способная выполнить действия, предписываемые алгоритмом.

**Алгоритмизация** — описание очередности выполнения различных операций, необходимых для решения той или иной задачи в форме алгоритма.

# 20. Объясните поэтапную схему решения задачи на ЭВМ.

Эвм - это исполнитель алгоритма

**Основные этапы решения задач на ЭВМ:**

1. Построение задачи.
2. Формирование задачи.
3. Построение алгоритма.
4. Составление программы на ЯП.
5. Отладка и тестирование.
6. Проведение расчетов и анализ полученных результатов.

# 21. Объясните принципы деления данных.

**Данные -** любая информация, представленная в сформулированном виде и пригодная для обработки алгоритма.

Есть 2 вида данных

1. **Искомые -** известные перед началом алготорма( Pi = 3.14)
2. **Конечные (выходные) -** данные являющиеся результатом действия алгоритма.

Так же данные могут делаться на

1. **Переменная -** данными можно крутить как угодно
2. **Константа** - значение которое устанавливается и не может меняется в процессе выполнения
3. **Идентификатор -** Последовательность букв и цифр (начало с буквы)

# 22. Объясните принципы использования операторов перехода.

**GoTo(**также оператор безусловного перехода**)** (переходит к определенной строке (может выполняться бесконечно))

# 23. Объясните причины ошибок при программировании циклов.

*1.Неверно указан тип переменных.*

**var** x: **real**;

**begin**

**for** x := 1 **to** 10 **do**

*2****.*** *Использование одинаковой переменной цикла в заголовках вложенных циклов*

**var** i,j: **integer**;

**begin**

**for** i := 1 **to** 10 **do**

**for** i := 1 **to** 10 **do**

*3. Не задали начало переменной*

**var**

i: **integer**;

p: **real**;

**begin**

*// Забыли инициализировать p := 1*

**for** i := 1 **to** 10 **do**

p := p \* 2;

*4. Зацикливание*

**var** x: **integer**;

**begin**

x := 1;

**while** x>0 **do** *// ошибка: условие все время остается истинным*

**begin**

writeln(x);

x := x + 1;

## **24. Охарактеризуйте поколения языков программирования.**

Язы́к программи́рования — [формальный язык](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA), предназначенный для записи [компьютерных программ](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)

Существует несколько поколений:

1. Машинные
2. Ассемблеры
3. Высокоуровневые языки
4. Непроцедурные, Объектно-ориентированные, языки запросов, параллельные
5. Языки искусственного интеллекта, экспертных систем и баз знаний, естественные языки

## **25. Охарактеризуйте этапы решения задачи на ЭВМ.**

Эвм - это исполнитель алгоритма

**Основные этапы решения задач на ЭВМ:**

1.Построение задачи.

2.Формирование задачи.

3.Построение алгоритма.

4.Составление программы на ЯП.

5.Отладка и тестирование.

6.Проведение расчетов и анализ полученных результатов.

# 26. Перечислите основные свойства алгоритма и приведите примеры.

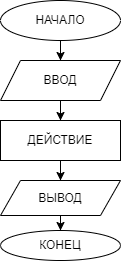
1. Дискретность (прерывность, раздельность) - Следующий шаг алгоритма выполняется после того, как выполнен предыдущий шаг.

Например, подъем по этажам. Не поднявшись на второй, не поднимешься на третий.

1. Понятность(определенность) - Каждое правило алгоритма должно быть четким, однозначным и не оставлять места для произвола. Например, инструкция на русском.
2. Результативность(конечность) - Алгоритм должен приводить к решению задачи за конечное число шагов. Например, алгоритм деления дробей.
3. Массовость - должен быть применимым для некоторого класса задач, различными только исходными данными. Например, алгоритм арифметических действий.
4. Детерминированность(однозначность) - каждая команда алгоритма определяет однозначность действий. Например, посадка и взлёт самолёта или покупка продуктов строго по списку.

# 27. Принципы построения линейной структуры (пример).

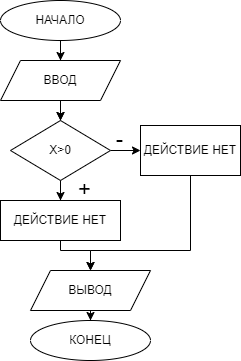
Линейный алгоритм - набор команд (указаний) выполняемых во времени последовательно друг за другом.



## 

# 28. Принципы построения структуры Ветвления (пример).

Разветвляющийся алгоритм - алгоритм, содержащий хотя бы одно условие, в результате проверки которого эвм обеспечивает переход на один из двух возможных шагов.



# 29. Принципы построения структуры Выбор (пример).

switch/case

## **30. Принципы построения структуры цикла Для (пример).**

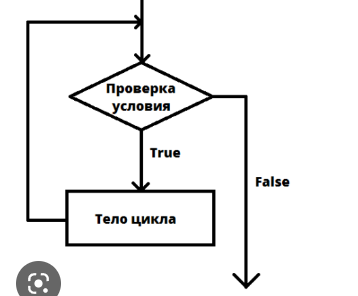
Цикл для зависит от контекста - для того чтобы F стала = 15 , надо сделать инкремент 5 раз.

Фиксированное количество повторов **For-** Цикл выполняется столько сколько ему задали

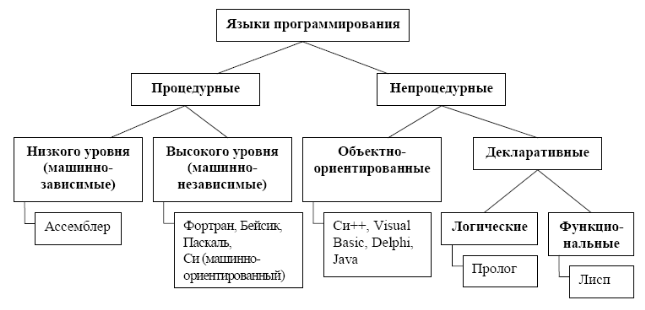


## **31. Принципы построения структуры цикла Пока (пример).**

Цикл Пока(While) - позволяет выполнить одну и ту же последовательность действий, пока проверяемое условие истинно.



## **32. Проведите классификацию языков программирования.**

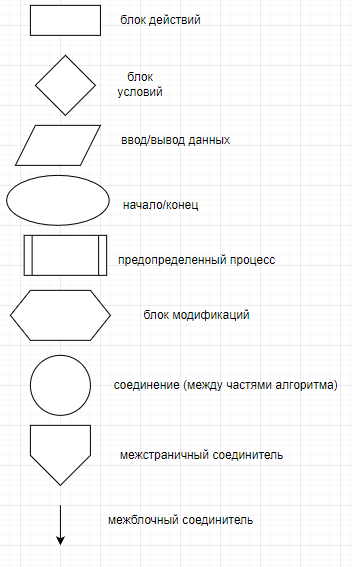


## 33. Разберите структуру исходной программы.

## 34. Расскажите об основных формах записи алгоритмов.

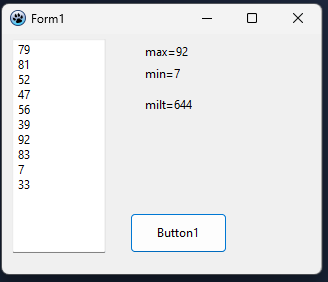
1. словесную форму (запись на естественном языке);
2. формульно-словесную форму (запись с помощью ограниченного набора синтаксических конструкций, на языке «псевдокод»);
3. графическую форму (запись на языке блок-схем);
4. программную форму (на языке исполнителя).

## 35. Рассказать о правилах построения блок-схем алгоритмов.

****

Практические задания

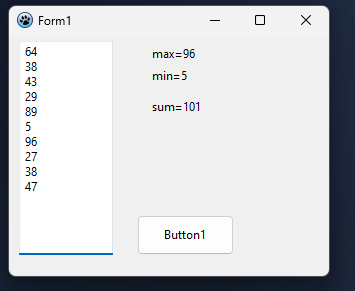
## **1. В заданном массиве *A(N)* найти произведение наибольшего и наименьшего элементов.**



| procedure TForm1.Button1Click(Sender: TObject); var a:array [1..10] of integer;  min,max,i:integer; begin Memo1.Lines.Clear ; i:=1;  repeat a[i]:=random(100); // (100) тут задается максимум в рандоме Memo1.Lines.Add(FloatToStr(a[i])); if i=1 then  begin  max := a[i];  min:=a[i];  i:=i+1;  continue; //закончит текущий повтор цикла  end; if a[i]>max then  begin  Max := a[i] ;  end; if a[i]<min then  begin  min := a[i] ;  end; i:=i+1; until i>10 ;  Label1.caption:= 'max='+floattostr(max); Label2.caption:= 'min='+floattostr(min) ; Label3.caption:= 'milt='+floattostr(max\*min) end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random= new Random();  int[] MyArray = new int[10];   for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i]= random.Next(10);  Console.WriteLine($"{i+1} элемент массива: {MyArray[i]}");  }   Console.WriteLine($"\nПроизведение min & max = {MyArray.Max()\*MyArray.Min()}");  }  }  internal class Program  {  static void Main(string[] args)  {  Random random= new Random();  int[] MyArray = new int[10];  for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i]= random.Next(1,100);  Console.WriteLine($"{i+1} элемент массива: {MyArray[i]}");  }    Console.WriteLine($"\nMin:{MyArray.Min()} Max:{MyArray.Max()}");  Console.WriteLine($"Произведение min & max = {MyArray.Max()\*MyArray.Min()}");  }  } |
| --- |

## **2. В заданном массиве *A(N)* найти сумму наибольшего и наименьшего элементов.**



| procedure TForm1.Button1Click(Sender: TObject); var a:array [1..10] of integer;  min,max,i:integer; begin Memo1.Lines.Clear ; i:=1;  repeat a[i]:=random(100); // (100) тут задается максимум в рандоме Memo1.Lines.Add(FloatToStr(a[i])); if i=1 then  begin  max := a[i];  min:=a[i];  i:=i+1;  continue; //закончит текущий повтор цикла  end; if a[i]>max then  begin  Max := a[i] ;  end; if a[i]<min then  begin  min := a[i] ;  end; i:=i+1; until i>10 ;  Label1.caption:= 'max='+floattostr(max); Label2.caption:= 'min='+floattostr(min) ; Label3.caption:= 'sum='+floattostr(max+min)  end; |
| --- |

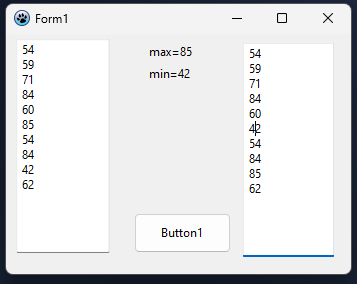
## 

| static void Main(string[] args)  {  Random random= new Random();  int[] MyArray = new int[10];   for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i]= random.Next(10);  Console.WriteLine($"{i+1} элемент массива: {MyArray[i]}");  }   Console.WriteLine($"\nСумма min & max = {MyArray.Max()+MyArray.Min()}");  } |
| --- |

## 

## 

## **3. В заданном массиве A(N) поменяйте местами наибольший и наименьший элементы**



| procedure TForm1.Button1Click(Sender: TObject); var  a:array [1..10] of integer;  max, min, IndMin, IndMax, i:integer; begin Memo1.Lines.Clear ; //очистка мемо Memo2.Lines.Clear ;  for i:=1 to 10 do  begin  a[i]:=random(100); // (100) тут задается максимум в рандоме Memo1.Lines.Add(FloatToStr(a[i]));  if i=1 then  begin  max:=a[i];  min:=a[i];  IndMax:=i;  IndMin:=i;  continue; //закончит текущий повтор цикла  end;  if a[i]>max then  begin  max:=a[i];  indMax:=i;  end;  if a[i]<min then  begin  min:=a[i];  IndMin:=i;  end;  end;  a[IndMax]:= min; a[IndMin]:= max; Label1.caption:= 'max='+floattostr(max); Label2.caption:= 'min='+floattostr(min) ;  for i:=1 to 10 do  begin  Memo2.Lines.Add(FloatToStr(a[i]));  end;  end; |
| --- |

## 

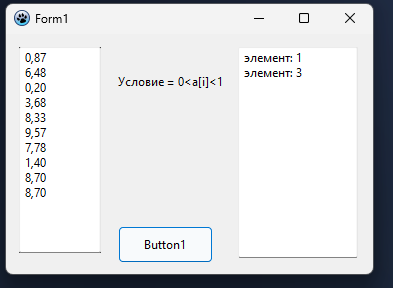


| static void Main(string[] args)  {  Random random= new Random();  int[] MyArray = new int[10];   Console.WriteLine("\t 1 массив");  for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i]= random.Next(100);  Console.WriteLine($"{i+1} элемент массива: {MyArray[i]}");  }  Console.WriteLine("\n");   int maxInd = Array.IndexOf(MyArray, MyArray.Max());   int minInd = Array.IndexOf(MyArray, MyArray.Min());   int max = MyArray.Max();  int min = MyArray.Min();   Console.WriteLine($"Min[{minInd + 1}]:{MyArray.Min()} Max[{maxInd + 1}]:{MyArray.Max()}");   MyArray[maxInd] = min;  MyArray[minInd] = max;   Console.WriteLine("\n\t 2 массив");  for (int i = 0; i < MyArray.Length; i++)  {  Console.WriteLine($"{i + 1} элемент массива: {MyArray[i]}");  }  } |
| --- |

## 

## 

## **4. Выведите на печать номера элементов заданного массива Y(N), удовлетворяющих условию 0<Y[i]<1**



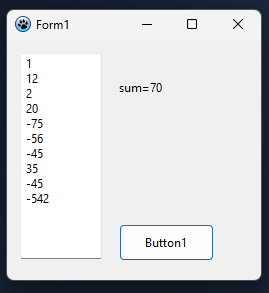
| procedure TForm1.Button1Click(Sender: TObject); var  a:array [1..10] of real;  i:integer; begin Memo1.Lines.Clear ; // очистка мемо Memo2.Lines.Clear ;  for i:=1 to 10 do  begin  a[i]:=random\*10; // создает рандомные числа в диапазоне 0-1 Memo1.Lines.Add(FloatToStrF(a[i],ffFixed,0,2));// делает 2 знака после , end;   for i:=1 to 10 do  begin   if (0<a[i]) and (a[i]<1) then   Memo2.Lines.Add('элемент: '+FloatToStr(i))   end; end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  float[] MyArray = new float[10];   Console.WriteLine("Массив:");  for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = (float) Math.Round(random.NextDouble() \* 10 , 2);  Console.Write(MyArray[i]+"\t");  }   Console.WriteLine("\nЭлементы удовлетворяющие условие:");  for (int i = 0; i < MyArray.Length; i++)  {  if (0 < MyArray[i] && MyArray[i] < 1)  {  Console.WriteLine($"{i+1} элемент ");  }  }  Console.WriteLine();  } |
| --- |

## **5. Дан массив A(N). Получите массив B(N), i-й элемент которого равен среднему арифметическому первых i элементов массива.**

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArrayOld = new int[5];  float[] MyArrayNew = new float[6];  float SredAr = 0;   Console.WriteLine("\t1 Массив\n");  for (int i = 0; i < MyArrayOld.GetLength(0); i++)  {  MyArrayOld[i] = random.Next(3);  Console.Write(MyArrayOld[i] + "\t");  SredAr += MyArrayOld[i];  }  SredAr /= MyArrayOld.GetLength(0);  Console.WriteLine($"\n\nCреднеe арифметическоe: {SredAr}\n");   int j = 0;  Console.WriteLine("\t2 Массив\n");  for (int i = 0; i < MyArrayNew.Length; i++)  {  if (i==0)  {  MyArrayNew[i]= SredAr;  Console.Write(MyArrayNew[i] + "\t");  continue;  }   MyArrayNew[i]= MyArrayOld[j];  Console.Write(MyArrayNew[i] + "\t");  j++;  }   } |
| --- |

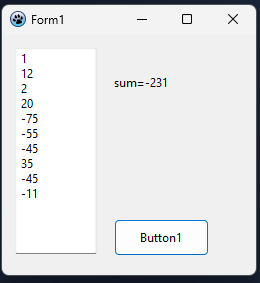
## **6. Дан целочисленный массив А размера N. Вывести сумму положительных элементов заданного массива A(N).**



| procedure TForm1.Button1Click(Sender: TObject); var  a:array [1..10]of integer;  sum,i:integer; begin Memo1.Lines.Clear ; // очистка мемо sum:=0; a[1]:=1 ; a[2]:=12 ; a[3]:=2 ; a[4]:=20 ; a[5]:=-75 ; a[6]:=-56 ; a[7]:=-45 ; a[8]:=35 ; a[9]:=-45 ; a[10]:=-542 ;  for i:=1 to 10 do  begin Memo1.Lines.Add(FloatToStr(a[i])); if a[i]>=0 then  begin  sum:= a[i]+sum ;  end; end; Label1.Caption:= 'sum='+ floattostr(sum); end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];  int sum = 0;  for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(-5, 5);  Console.Write(MyArray[i]+"\t");  if (MyArray[i]>0)  {  sum += MyArray[i];  }  }   Console.WriteLine($"\nСумма Положительных элементов = {sum}");  }  } |
| --- |

## **7. Дан целочисленный массив А размера N. Вывести сумму отрицательных элементов заданного массива A(N).**



| procedure TForm1.Button1Click(Sender: TObject); var  a:array [1..10]of integer;  sum,i:integer; begin Memo1.Lines.Clear ; // очистка мемо sum:=0; a[1]:=1 ; a[2]:= 12 ; a[3]:= 2 ; a[4]:= 20 ; a[5]:= -75 ; a[6]:= -55 ; a[7]:= -45 ; a[8]:= 35 ; a[9]:= -45 ; a[10]:= -11 ;  for i:=1 to 10 do  begin Memo1.Lines.Add(FloatToStr(a[i])); if a[i]<=0 then  begin  sum:= a[i]+sum ;  end; end; Label1.Caption:= 'sum='+ floattostr(sum); end; |
| --- |

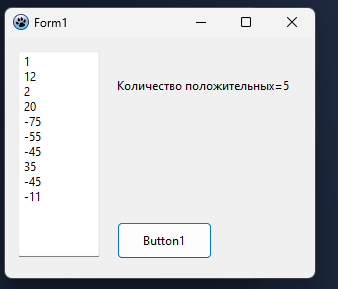
| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];  int sum = 0;  for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(-5, 5);  Console.Write(MyArray[i]+"\t");  if (MyArray[i]<0)  {  sum += MyArray[i];  }  }   Console.WriteLine($"\nСумма отрицательных элементов = {sum}");  }  } |
| --- |

## **8. Дан целочисленный массив А размера N. Вывести число отрицательных элементов заданного массива A(N).**

| procedure TForm1.Button1Click(Sender: TObject); var  a:array [1..10]of integer;  numberNeg,i:integer; begin Memo1.Lines.Clear ; // очистка мемо numberNeg:=0; a[1]:=1 ; a[2]:= 12 ; a[3]:= 2 ; a[4]:= 20 ; a[5]:= -75 ; a[6]:= -55 ; a[7]:= -45 ; a[8]:= 35 ; a[9]:= -45 ; a[10]:= -11 ;  for i:=1 to 10 do  begin Memo1.Lines.Add(FloatToStr(a[i])); if a[i]<=0 then  begin  numberNeg:= numberNeg+1 ;  end; end; Label1.Caption:= 'количество отрицательных ='+ floattostr(numberNeg); end; |
| --- |

| static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];  int value = 0;  for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(-5, 5);  Console.Write(MyArray[i]+"\t");  if (MyArray[i]<0)  {  value++;  }  }   Console.WriteLine($"\nКоличество Эллементов 0 < : {value}");  } |
| --- |

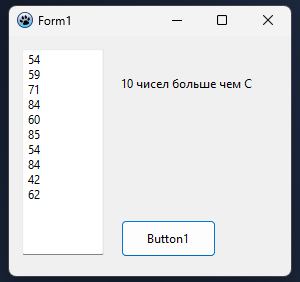
## **9. Дан целочисленный массив А размера N. Вывести число положительных элементов заданного массива A(N).**



| procedure TForm1.Button1Click(Sender: TObject); var  a:array [1..10]of integer;  numberPos,i:integer; begin Memo1.Lines.Clear ; // очистка мемо numberPos:=0; a[1]:=1 ; a[2]:= 12 ; a[3]:= 2 ; a[4]:= 20 ; a[5]:= -75 ; a[6]:= -55 ; a[7]:= -45 ; a[8]:= 35 ; a[9]:= -45 ; a[10]:= -11 ;  for i:=1 to 10 do  begin Memo1.Lines.Add(FloatToStr(a[i])); if a[i]>=0 then  begin  numberPos:= numberPos+1 ;  end; end; Label1.Caption:= 'Количество положительных='+ floattostr(numberPos); end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];  int value = 0;  for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(-5, 5);  Console.Write(MyArray[i]+"\t");  if (MyArray[i]>0)  {  value++;  }  }   Console.WriteLine($"\nКоличество Эллементов 0 > : {value}");  }  } |
| --- |

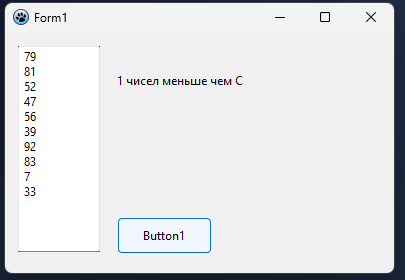
## **10. Дан целочисленный массив А размера N. Найти количество элементов массива А, удовлетворяющих условию с<=a[i].**



| procedure TForm1.Button1Click(Sender: TObject); var  a:array [1..10]of integer;  numSuittable,c,i:integer; begin Memo1.Lines.Clear ; // очистка мемо c:=25; //c - любое число ,я возьму его как 25 numSuittable:=0;// подходящие нам числа под условие for i:=1 to 10 do  begin   a[i]:=random(100);  Memo1.Lines.Add(FloatToStr(a[i]));  if c<=a[i] then  begin  numSuittable:= numSuittable+1;  end;  end; Label1.caption:= floattostr(numSuittable) + ' чисел больше чем C'; end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];  Console.Write("Введите число С:");  float c = float.Parse(Console.ReadLine());  int condition = 0;   for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(0, 50);  Console.Write(MyArray[i]+"\t");  if (MyArray[i]>=c)  {  condition++;  }  }   Console.WriteLine($"\n\n{condition} болше число С");  }  } |
| --- |

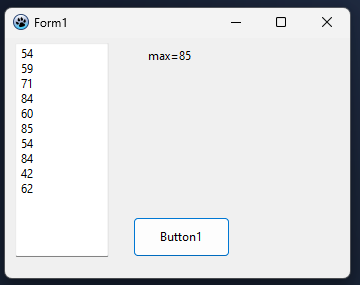
## **11. Дан целочисленный массив А размера N. Найти количество элементов массива А, удовлетворяющих условию с>a[i].**



| procedure TForm1.Button1Click(Sender: TObject); var  a:array [1..10]of integer;  numSuittable,c,i:integer; begin Memo1.Lines.Clear ; // очистка мемо c:=25; //c - любое число ,я возьму его как 25 numSuittable:=0;// подходящие нам числа под условие for i:=1 to 10 do  begin   a[i]:=random(100);  Memo1.Lines.Add(FloatToStr(a[i]));  if c>=a[i] then  begin  numSuittable:= numSuittable+1;  end;  end; Label1.caption:= floattostr(numSuittable) + ' чисел меньше чем C'; end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];  Console.Write("Введите число С:");  float c = float.Parse(Console.ReadLine());  int condition = 0;   for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(0, 50);  Console.Write(MyArray[i]+"\t");  if (MyArray[i]<c)  {  condition++;  }  }   Console.WriteLine($"\n\n{condition} меньше С");  }  } |
| --- |

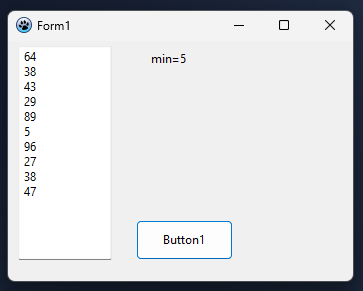
## **12. Дан целочисленный массив А размера N. Найти максимальный элемент массива.**



| procedure TForm1.Button1Click(Sender: TObject); var a:array [1..10] of integer;  max,i:integer; begin i:=1; Memo1.Lines.Clear ; // очистка мемо  repeat a[i]:=random(100); // (100) тут задается максимум в рандоме Memo1.Lines.Add(FloatToStr(a[i]));  if i=1 then  begin  max := a[i];  i:=i+1;  continue; //закончит текущий повтор цикла   end; if a[i]>max then  begin  max := a[i] ;  end;  i:=i+1; until i>10 ;  Label1.caption:= 'max='+floattostr(max); end; |
| --- |

| **static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];    for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(0, 50);  Console.Write(MyArray[i]+"\t");  }   Console.WriteLine($"\n\n{MyArray.Max()}: самое большое число в массиве");  }** |
| --- |

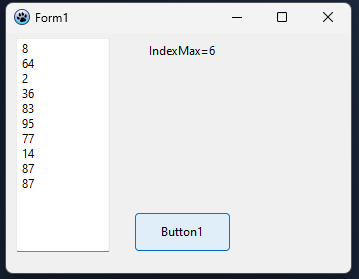
## **13. Дан целочисленный массив А размера N. Найти минимальный элемент массива.**



| procedure TForm1.Button1Click(Sender: TObject); var a:array [1..10] of integer;  min,i:integer; begin i:=1; Memo1.Lines.Clear ; // очистка мемо  repeat a[i]:=random(100); // (100) тут задается максимум в рандоме Memo1.Lines.Add(FloatToStr(a[i]));  if i=1 then  begin  min := a[i];  i:=i+1;  continue; //закончит текущий повтор цикла   end; if a[i]<min then  begin  min := a[i] ;  end;  i:=i+1; until i>10 ;  Label1.caption:= 'min='+floattostr(min); end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];    for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(0, 50);  Console.Write(MyArray[i]+"\t");  }   Console.WriteLine($"\n\n{MyArray.Min()}: самое маленькое число в массиве");  }  } |
| --- |

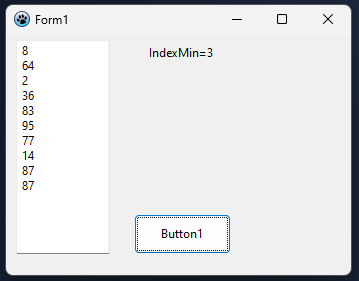
## **14. Дан целочисленный массив А размера N. Найти номер максимального элемента массива.**



| procedure TForm1.Button1Click(Sender: TObject); var a:array [1..10] of integer;  IndexMax,max,i:integer; begin i:=1; Memo1.Lines.Clear ; // очистка мемо  repeat a[i]:=random(100); Memo1.Lines.Add(FloatToStr(a[i]));  if i=1 then  begin  max := a[i];  IndexMax := i;  i:=i+1;  continue;  end; if a[i]>max then  begin  max := a[i] ;  IndexMax := i;  end;  i:=i+1; until i>10 ;  Label1.caption:= 'IndexMax='+floattostr(IndexMax); end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];    for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(0, 10);  Console.Write(MyArray[i]+"\t");  }   int maxInd = Array.IndexOf(MyArray, MyArray.Max());  Console.WriteLine($"\n\nЧисло под индексом {maxInd+1} масое большое= {MyArray.Max()}");  }  } |
| --- |

## **15. Дан целочисленный массив А размера N. Найти номер минимального элемента массива.**

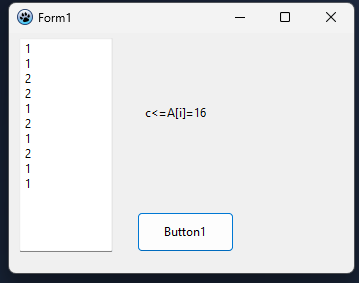


| procedure TForm1.Button1Click(Sender: TObject); var a:array [1..10] of integer;  IndexMin,min,i:integer; begin i:=1; Memo1.Lines.Clear ; // очистка мемо  repeat a[i]:=random(100); Memo1.Lines.Add(FloatToStr(a[i]));  if i=1 then  begin  min := a[i];  IndexMin := i;  i:=i+1;  continue;  end; if a[i]<min then  begin  min := a[i] ;  IndexMin := i;  end;  i:=i+1; until i>10 ;  Label1.caption:= 'IndexMin='+floattostr(IndexMin); end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];    for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(0, 10);  Console.Write(MyArray[i]+"\t");  }   int minInd = Array.IndexOf(MyArray, MyArray.Min());  Console.WriteLine($"\n\nЧисло под индексом {minInd+1} масое маленькое= {MyArray.Min()}");  }  } |
| --- |

## 

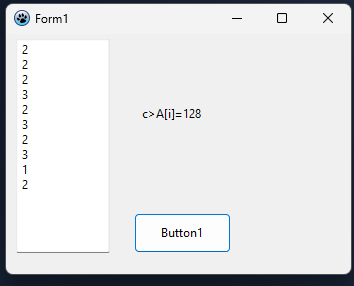
## **16. Дан целочисленный массив А размера N. Найти произведение элементов массива А, удовлетворяющих условию с<=А[i].**



| procedure TForm1.Button1Click(Sender: TObject); const  c: integer= 1; var a:array [1..10] of integer;  result,i:integer; begin result :=0; i:=1; Memo1.Lines.Clear ; // очистка мемо  repeat a[i]:=random(3); Memo1.Lines.Add(FloatToStr(a[i]));  if c <= a[i] then  begin  if result = 0 then   begin  result:= a[i] ;  end;  result:=result\*a[i] ;  end;  i:=i+1; until i>10 ;  Label1.caption:= 'с<=А[i]='+floattostr(result); end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];  Console.Write("Введите число С:");  int c = int.Parse(Console.ReadLine());  int result = 1;   for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(0, 10);  Console.Write(MyArray[i]+"\t");  if (MyArray[i]>=c)  {  result \*= MyArray[i];  }  }  Console.WriteLine($"\n\nПроизведение всех чисел которые > | = С Равно:{result}");  }  } |
| --- |

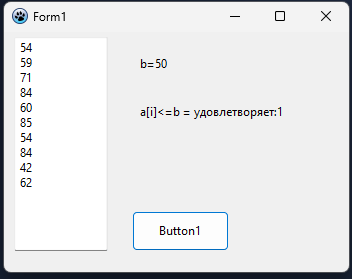
## **17. Дан целочисленный массив А размера N. Найти произведение элементов массива А, удовлетворяющих условию с>a[i].**



| procedure TForm1.Button1Click(Sender: TObject); const  c: integer= 3; var a:array [1..10] of integer;  result,i:integer; begin result :=0; i:=1; Memo1.Lines.Clear ; // очистка мемо  repeat a[i]:=random(4); Memo1.Lines.Add(FloatToStr(a[i]));  if c > a[i] then  begin  if result = 0 then  begin  result:= a[i] ;  end;  result:=result\*a[i] ;  end;  i:=i+1; until i>10 ;  Label1.caption:= 'с>А[i]='+floattostr(result); end; |
| --- |

| **internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];  Console.Write("Введите число С:");  int c = int.Parse(Console.ReadLine());  int result = 1;   for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(1, 10);  Console.Write(MyArray[i]+"\t");  if (MyArray[i]<c)  {  result \*= MyArray[i];  }  }  Console.WriteLine($"\n\nПроизведение всех чисел которые < С Равно:{result}");  }  }** |
| --- |

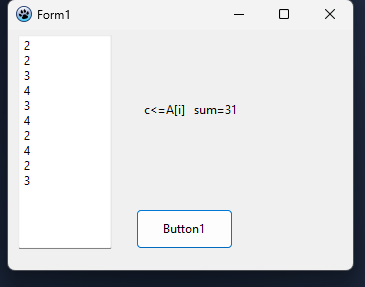
## **18. Дан целочисленный массив А размера N. Найти сколько элементов массива А удовлетворяют условию: a[i]<=d**



| procedure TForm1.Button1Click(Sender: TObject); const  b: integer= 50; var a:array [1..10] of integer;  index,i:integer; begin index :=0; i:=1; Memo1.Lines.Clear ; // очистка мемо  repeat a[i]:=random(100); Memo1.Lines.Add(FloatToStr(a[i]));  if a[i]<=b then  begin  index :=index+1;  end;  i:=i+1; until i>10 ;  Label1.caption:= 'a[i]<=b = удовлетворяет:'+floattostr(index); end; |
| --- |

| **internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];  Console.Write("Введите число d:");  int d = int.Parse(Console.ReadLine());  int result = 0;   for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(1, 10);  Console.Write(MyArray[i]+"\t");  if (MyArray[i]<=d)  {  result++;  }  }  Console.WriteLine($"\n\n{result} элементов удовлетворяет условиею a[i]<=d");  }  }** |
| --- |

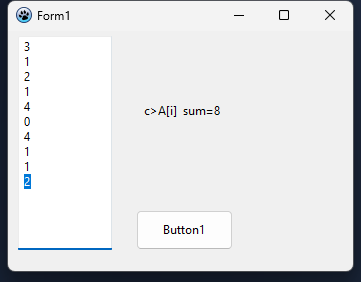
## **19. Дан целочисленный массив А размера N. Найти сумму элементов массива А, удовлетворяющих условию с<=a[i].**



| procedure TForm1.Button1Click(Sender: TObject); const  c: integer= 2; var a:array [1..10] of integer;  result,i:integer; begin result :=0; i:=1; Memo1.Lines.Clear ; // очистка мемо  repeat a[i]:=random(5); Memo1.Lines.Add(FloatToStr(a[i]));  if c <= a[i] then  begin  result:=result+a[i] ;  end;  i:=i+1; until i>10 ;  Label1.caption:= 'с<=А[i] sum='+floattostr(result); end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];  Console.Write("Введите число с:");  int с = int.Parse(Console.ReadLine());  int result = 0;   for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(0, 10);  Console.Write(MyArray[i]+"\t");  if (MyArray[i]<=с)  {  result+= MyArray[i];  }  }  Console.WriteLine($"\n\nСумма элементов удовлетворяет условиею a[i]>=с = {result}");  }  } |
| --- |

## **20. Дан целочисленный массив А размера N. Найти сумму элементов массива А, удовлетворяющих условию с>a[i].**

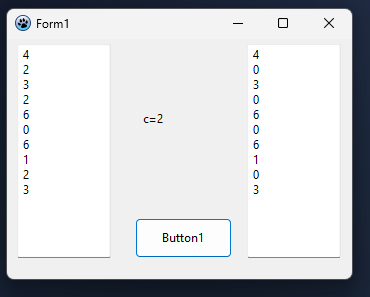


| procedure TForm1.Button1Click(Sender: TObject); const  c: integer= 3; var a:array [1..10] of integer;  result,i:integer; begin result :=0; i:=1; Memo1.Lines.Clear ; // очистка мемо  repeat a[i]:=random(5); Memo1.Lines.Add(FloatToStr(a[i]));  if c > a[i] then  begin  result:=result+a[i] ;  end;  i:=i+1; until i>10 ;  Label1.caption:= 'с>А[i] sum='+floattostr(result); end; |
| --- |

\

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];  Console.Write("Введите число с:");  int с = int.Parse(Console.ReadLine());  int result = 0;   for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(0, 10);  Console.Write(MyArray[i]+"\t");  if (MyArray[i]<с)  {  result+= MyArray[i];  }  }  Console.WriteLine($"\n\nСумма элементов удовлетворяет условиею a[i]<с = {result}");  }  } |
| --- |

## **21. Дано число с – целое и массив целых чисел А из 10 элементов. В заданном массиве все элементы со значением с сделать равными 0.**



| procedure TForm1.Button1Click(Sender: TObject); const  c: integer= 2; var a:array [1..10] of integer;  result,i:integer; begin result :=0; i:=1; Memo1.Lines.Clear ; // очистка мемо Memo2.Lines.Clear ;  repeat a[i]:=random(7); Memo1.Lines.Add(FloatToStr(a[i]));  if c = a[i] then  begin  a[i]:=0;  end; Memo2.Lines.Add(FloatToStr(a[i])); i:=i+1; until i>10 ;  Label1.caption:= 'с=2';  end; |
| --- |

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();  int[] MyArray = new int[10];   Console.WriteLine("1 Массив");  for (int i = 0; i < MyArray.Length; i++)  {  MyArray[i] = random.Next(0, 10);  Console.Write(MyArray[i] + "\t");  }   Console.Write("\n\nВведите число с:");  int с = int.Parse(Console.ReadLine());   Console.WriteLine("\n2 Массив");  for (int i = 0; i < MyArray.Length; i++)  {  if (MyArray[i]==с)  {  MyArray[i] = 0;  }  Console.Write(MyArray[i] + "\t");  }  }  } |
| --- |

## **22. Задана квадратная матрица А размером NxN (N<=10), состоящая из вещественных чисел. Найти среднее арифметическое элементов каждой из строк этой матрицы.**

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();    float[,] MyArray = new float[5 ,5];   for (int i = 0; i < MyArray.GetLength(0); i++)  {  for (int j = 0; j < MyArray.GetLength(1); j++)  {  MyArray[i, j] = (float)random.Next(-1000,1000)/10;  Console.Write($"{MyArray[i, j]} \t");  }  Console.WriteLine();  }   Console.WriteLine("\n\n\n");  float SredAr=0;   for (int i = 0; i < MyArray.GetLength(0); i++)  {  for (int j = 0; j < MyArray.GetLength(1); j++)  {  SredAr += MyArray[i, j];  }  Console.WriteLine($"{i+1} строка: "+ Math.Round((float)SredAr/MyArray.GetLength(1), 2));  SredAr = 0;  }  }  } |
| --- |

## 

## **23. Задана квадратная матрица А размером NxN (N<=10), состоящая из вещественных чисел. Найти среднее арифметическое элементов каждого из столбцов этой матрицы**

| internal class Program  {  static void Main(string[] args)  {  Random random = new Random();    float[,] MyArray = new float[5 ,5];   for (int i = 0; i < MyArray.GetLength(0); i++)  {  for (int j = 0; j < MyArray.GetLength(1); j++)  {  MyArray[i, j] = (float)random.Next(-1000,1000)/10;  Console.Write($"{MyArray[i, j]} \t");  }  Console.WriteLine();  }   Console.WriteLine("\n\n\n");  float SredAr=0;   for (int i = 0; i < MyArray.GetLength(1); i++)  {  for (int j = 0; j < MyArray.GetLength(0); j++)  {  SredAr += MyArray[j, i];  }  Console.WriteLine($"{i+1} колонка: "+ Math.Round((float)SredAr/MyArray.GetLength(0), 2));  SredAr = 0;  }  }  } |
| --- |

## **24. Задана квадратная матрица А размером NxN (N<=10), состоящая из вещественных чисел. Найти сумму каждого из столбцов этой матрицы.**

| static void Main(string[] args)  {  Random random = new Random();   float[,] MyArray = new float[5, 5];   for (int i = 0; i < MyArray.GetLength(0); i++)  {  for (int j = 0; j < MyArray.GetLength(1); j++)  {  MyArray[i, j] = (float)random.Next(-1000, 1000) / 10;  Console.Write($"{MyArray[i, j]} \t");  }  Console.WriteLine();  }   Console.WriteLine("\n\n\n");  float result = 0;   for (int i = 0; i < MyArray.GetLength(1); i++)  {  for (int j = 0; j < MyArray.GetLength(0); j++)  {  result += MyArray[j, i];  }  Console.WriteLine($"{i + 1} колонка: {Math.Round(result, 1)}");  result = 0;  }  } |
| --- |

## **25. Задана квадратная матрица А размером NxN (N<=10), состоящая из вещественных чисел. Найти произведение каждого из столбцов этой матрицы**

| static void Main(string[] args)  {  Random random = new Random();   float[,] MyArray = new float[5, 5];   for (int i = 0; i < MyArray.GetLength(0); i++)  {  for (int j = 0; j < MyArray.GetLength(1); j++)  {  MyArray[i, j] = (float)random.Next(-100, 100) / 10;  Console.Write($"{MyArray[i, j]} \t");  }  Console.WriteLine();  }   Console.WriteLine("\n\n\n");  float result=1;   for (int i = 0; i < MyArray.GetLength(1); i++)  {  for (int j = 0; j < MyArray.GetLength(0); j++)  {  result \*= MyArray[j, i];  }  Console.WriteLine($"{i + 1} колонка: {Math.Round(result,2)}");  result = 1;  }  } |
| --- |

## **26. Задана квадратная матрица А размером NxN (N<=10), состоящая из вещественных чисел. Найти произведение каждой строки этой матрицы.**

| static void Main(string[] args)  {  Random random = new Random();   float[,] MyArray = new float[5, 5];   for (int i = 0; i < MyArray.GetLength(0); i++)  {  for (int j = 0; j < MyArray.GetLength(1); j++)  {  MyArray[i, j] = (float)random.Next(100) / 10;  Console.Write($"{MyArray[i, j]} \t");  }  Console.WriteLine();  }   Console.WriteLine("\n\n\n");  float result = 1;   for (int i = 0; i < MyArray.GetLength(0); i++)  {  for (int j = 0; j < MyArray.GetLength(1); j++)  {  result \*= MyArray[i, j];  }  Console.WriteLine($"{i + 1} строка: " + Math.Round((float)result,2));  result = 1;  }  } |
| --- |

## **27. Задана квадратная матрица А размером NxN (N<=10), состоящая из вещественных чисел. Найти сумму каждой строки этой матрицы.**

| static void Main(string[] args)  {  Random random = new Random();   float[,] MyArray = new float[5, 5];   for (int i = 0; i < MyArray.GetLength(0); i++)  {  for (int j = 0; j < MyArray.GetLength(1); j++)  {  MyArray[i, j] = (float)random.Next(-1000,1000) / 10;  Console.Write($"{MyArray[i, j]} \t");  }  Console.WriteLine();  }   Console.WriteLine("\n\n\n");  float result = 0;   for (int i = 0; i < MyArray.GetLength(0); i++)  {  for (int j = 0; j < MyArray.GetLength(1); j++)  {  result += MyArray[i, j];  }  Console.WriteLine($"{i + 1} строка: " + Math.Round((float)result,2));  result = 0;  }  } |
| --- |

## **28. Запишите подряд в массив A(N) элементы заданного массива B(2N), стоящие на нечетных местах.**

| static void Main(string[] args)  {  int[] MyArrayOld = new int [10];   int[] MyArrayNew= new int [MyArrayOld.Length/2];   Random random = new Random();   int j = 0;  Console.WriteLine(" Массив 1 ");  for (int i = 0; i < MyArrayOld.Length; i++)  {  MyArrayOld[i] = random.Next(255);  if (i%2==0)  {  MyArrayNew[j++] = MyArrayOld[i];  }  Console.WriteLine($"{i+1} элемент массива= {MyArrayOld[i]}");  }  Console.WriteLine("\n \t Массив 2 ");   for (int i = 0; i < MyArrayNew.Length; i++)  {  Console.WriteLine($"\t{i+1} элемент массива= {MyArrayNew[i]}");  }  } |
| --- |

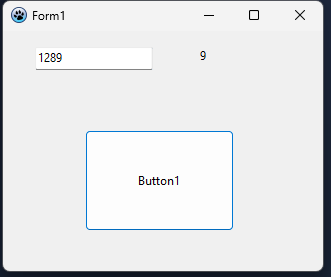
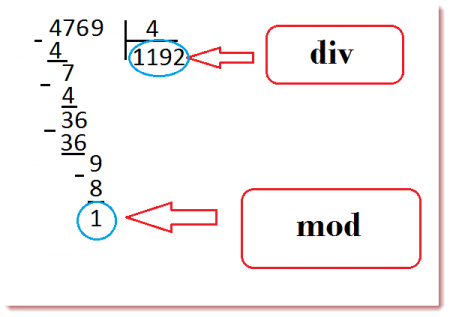
## **29. Запишите подряд в массив A(N) элементы заданного массива B(2N), стоящие на чётных местах.**

| static void Main(string[] args)  {  int[] MyArrayOld = new int [10];   int[] MyArrayNew= new int [MyArrayOld.Length/2];   Random random = new Random();   int j = 0;  Console.WriteLine(" Массив 1 ");  for (int i = 0; i < MyArrayOld.Length; i++)  {  MyArrayOld[i] = random.Next(255);  if (i%2==1)  {  MyArrayNew[j++] = MyArrayOld[i];  }  Console.WriteLine($"{i+1} элемент массива= {MyArrayOld[i]}");  }  Console.WriteLine("\n \t Массив 2 ");   for (int i = 0; i < MyArrayNew.Length; i++)  {  Console.WriteLine($"\t{i+1} элемент массива= {MyArrayNew[i]}");  }  } |
| --- |

## **30. Осуществите циклический сдвиг компонент заданного вектора A(N) влево на одну позицию.**

## **31. Осуществите циклический сдвиг компонент заданного вектора A(N) вправо на одну позицию.**

## **32. С клавиатуры вводится натуральное число. Найти его наибольшую цифру. Например, введено число 764580. Наибольшая цифра в нем 8.**



| procedure TForm1.Button1Click(Sender: TObject); var a,max:integer; begin a:= strtoint(edit1.Text); max:=0; while a>0 do begin  if a mod 10 > max then  begin  max:=a mod 10 ;  end;   a:=a div 10;  end;   label1.caption := floattostr(max) ; end ; |
| --- |

| static void Main(string[] args)  {  char [] MyArray;   MyArray =Console.ReadLine().ToCharArray();   Console.WriteLine(MyArray.Max());   } |
| --- |

## **33. Элементы заданного массива B(N) перепишите в новый массив A(N) в обратном порядке.**

## 

| procedure TForm1.Button1Click(Sender: TObject); var  a: array [1..5] of integer;  b: array [1..5] of integer;  i, j: integer; begin  Memo1.Lines.Clear;  Memo2.Lines.Clear; j:=5; i:=1 ; repeat  a[i]:=random(100);  Memo1.Lines.Add(floattostr(a[i]));  b[j]:=a[i];  i:=i+1;  j:=j-1; until i >5;  for j:=1 to 5 do begin  Memo2.Lines.Add(floattostr(b[j])) end ;  end; |
| --- |

## 

| static void Main(string[] args)  {  Random random= new Random();  int[] MyArrayOld = new int[10];  int[] MyArrayNew = new int[10];   Console.WriteLine("\t 1 массив");  for (int i = 0; i < MyArrayOld.Length; i++)  {  MyArrayOld[i] = random.Next(100);  Console.WriteLine($"{i+1} элемент массива: { MyArrayOld[i]}");  }  Console.WriteLine('\n');  int j = 9;   Console.WriteLine("\t 2 массив");  for (int i = 0; i < MyArrayNew.Length; i++)  {  MyArrayNew[i] = MyArrayOld[j--];  Console.WriteLine($"{i + 1} элемент массива: {MyArrayNew[i]}");  }  } |
| --- |

## 