

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Тульский государственный университет»

Институт горного дела и строительства  
Кафедра «Санитарно-технические системы»

Утверждено на заседании кафедры  
«Санитарно-технические системы»  
«20» января 2023 г., протокол № 5

Заведующий кафедрой

  
\_\_\_\_\_ Р.А. Ковалев

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ (ОЦЕНОЧНЫЕ МАТЕРИАЛЫ) ДЛЯ  
ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И  
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО  
ДИСЦИПЛИНЕ (МОДУЛЮ)**

**«Алгоритмизация и программирование»**

**основной профессиональной образовательной программы  
высшего образования – программы бакалавриата**

по направлению подготовки  
**08.03.01 – "Строительство"**

с направленностью (профилем)  
**"Водоснабжение и водоотведение"**

Форма(ы) обучения: очная, очно-заочная

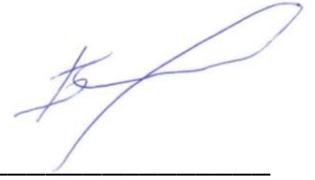
Идентификационный номер образовательной программы: 080301-02-23

Тула 2023 год

**ЛИСТ СОГЛАСОВАНИЯ**  
**фонда оценочных средств (оценочных материалов)**

**Разработчик(и):**

Белоусов Р.О., доцент, к.т.н., доцент  
(ФИО, должность, ученая степень, ученое звание)



---

(подпись)

## 1. Описание фонда оценочных средств (оценочных материалов)

Фонд оценочных средств (оценочные материалы) включает в себя контрольные задания и (или) вопросы, которые могут быть предложены обучающемуся в рамках текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю). Указанные контрольные задания и (или) вопросы позволяют оценить достижение обучающимся планируемых результатов обучения по дисциплине (модулю), установленных в соответствующей рабочей программе дисциплины (модуля), а также сформированность компетенций, установленных в соответствующей общей характеристике основной профессиональной образовательной программы.

Полные наименования компетенций и индикаторов их достижения представлены в общей характеристике основной профессиональной образовательной программы.

## 2. Оценочные средства (оценочные материалы) для проведения текущего контроля успеваемости обучающихся по дисциплине (модулю)

### 2 семестр

#### Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.1)

#### Построение заданного алгоритма из блоков

Необходимо указать порядок записи блоков построения алгоритма вычисления указанной величины для элементов таблицы  $A[1..N]$ .

1.

A	нц для k	B	нц для i
C	от 1 до N	D	от N до 1
E	если $0 > a[i]$	F	если $0 < a[k]$
G	то $s := a[k] + s$	H	все кц
I	то $s := s + a[i]$	J	$s := 0$

а) сумма положительных элементов: 1)ACFGH 2)BCFIH 3)JACFGH 4)BDFIH 5)JADEGH

б) сумма отрицательных элементов: 1)ACFGH 2)BCFIH 3)JBCFGH 4)BDFIH 5)JBCEIH.

2.

A	$s := 5$ ; нц для k	B	$s := 5$ ; нц для i
C	от 1 до N	D	от N до 1
E	если $(s > a[i])$	F	если $(s > a[k])$
G	то $s := a[k]$	H	все кц
I	то $s := a[i]$	J	или $(k = 1)$

минимальное значение в таблице: 1)BDFIH 2)BCFIH 3)ACFJGH 4)ADEGH 5)ADEJGH.

3.

A	нц для k	B	нц для i
C	от 1 до N	D	от N до 1 шаг -1
E	если ( $0 < a[k]$ )	F	$s := s + a[i]$
G	то $s := a[k]$ все	H	кц
I	$s := s/N$	J	$s := 0$

среднее арифметическое значений элементов: 1)JACEGHI 2)JBDFHI 3)JBDFIH 4)BDEGHI  
5)JBCFIH

4.

A	нц для k	B	нц для i
C	от 1 до N	D	от N до 1
E	если ( $x = a[k]$ )	F	если ( $x = a[i]$ )
G	то $s := a[k]$	H	все кц
I	то $p := i$	J	$p := 0;$

минимальный индекс элемента, равного x: 1)ACEGH 2)JBCFIH 3)BCFIH 4)JBDFIH  
5)JBDEGH

5.

A	нц для k	B	нц для i
C	от 1 до N	D	от N до 1
E	если ( $x = a[k]$ )	F	если ( $x = a[i]$ )
G	то $s := a[i]$	H	все кц
I	то $p := k$	J	$p := 0;$

максимальный индекс элемента, равного x: 1)JACEIH 2)BCFGH 3)BDFGH 4)ADEIH  
5)JADEIH

6.

A	нц для k	B	нц для i
C	от 1 до N	D	от N до 1
E	если $\text{mod}(k, 2) = 0$	F	если $\text{mod}(a[i], 2) = 0$
G	то $s := s + a[k]$	H	все кц
I	то $s := s + a[i]$	J	$s := 0;$

сумма элементов с четными индексами: 1)JBDFIH 2)BCFIH 3)BDFIH 4)JADEGH 5)ADEJGH

7.

A	нц для k	B	нц для i
---	----------	---	----------

C	от 1 до N	D	от N до 1 шаг -1
E	если ( $0 < a[k]$ )	F	$s := s + a[i]$
G	то $s := a[k]$ все	H	кц
I	$s := s/N$	J	$s := 0$

среднее арифметическое значений элементов: 1)ACEJGH 2)JACEGH 3)JBDFIH 4)ADEGH  
5)ACEGH

8.

A	$s := 1$ ; нц для k	B	$s := 1$ ; нц для i
C	от 1 до N	D	от N до 1
E	если ( $s < a[k]$ )	F	если ( $s < a[i]$ )
G	то $s := a[k]$	H	все кц
I	то $s := a[i]$	J	или ( $k=1$ )

максимальное значение в таблице: 1)ACEJGH 2)BCFIH 3)BDFIH 4)ADEGH 5)ADEJGH.

### Определить вид действия, выполняемый алгоритмом

1. Задан одномерный массив  $x[1..N]$ . Фрагмент алгоритма

```
s:=0; нц для k от 1 до N
| если (k=1) или (s>x[k])
| | то s:= x[k]
| все
| кц
```

определяет:

- 1) минимальный элемент массива; 2) сумму отрицательных элементов;
- 3) количество отрицательных элементов; 4) индекс последнего отрицательного элемента;
- 5) максимальный элемент.

2. Задан двумерный массив  $x[1..n, 1..n]$ . Фрагмент алгоритма

```
нц для i от 1 до n
| нц для j от 1 до n
| | c:=a[i,j]; a[i,j]:=a[n-i+1,j]; a[n-i+1,j]:=c;
| кц
| кц
```

- 1) меняет порядок строк; 2) меняет порядок столбцов;
- 3) меняет местами элементы главной и побочной диагонали;
- 4) меняет местами элементы i-ой строки и j-го столбца; 5) ничего не меняет в таблице.

3. Задан одномерный массив  $x[1..N]$ . Фрагмент алгоритма

```
s:=0; нц для k от 1 до N
| если ( $0 < x[k]$ )
| | то  $s := s + x[k]$ 
| все
| кц
```

определяет:

- 1) максимальный элемент массива; 2) сумму положительных элементов;
- 3) количество положительных элементов; 4) индекс последнего положительного элемента;
- 5) индекс первого положительного элемента.

4. Задан одномерный массив  $x[1..N]$ . Фрагмент алгоритма

```
T:=0; нц для k от 1 до N
| если (k=1) или (T>x[k])
|| то T:=x[k]
| все
кц
```

определяет:

- 1) минимальный элемент массива; 2) сумму элементов массива, меньших T;
- 3) количество элементов, меньших T; 4) индекс последнего из элементов, меньших T;
- 5) индекс первого из элементов, меньших T.

5. Задан двумерный массив  $x[1..n,1..n]$ . В результате работы фрагмента алгоритма

```
нц для i от 1 до n
| нц для j от 1 до n
|| c:=a[i,j]; a[i,j]:=a[j,i]; a[j,i]:=c;
| кц
кц
```

значения переменных будут соответственно равны

- 1)  $a_{31}=13, a_{23}=4$ ; 2)  $a_{31}=14, a_{23}=0$ ; 3)  $a_{31}=12, a_{23}=11$ ; 4)  $a_{31}=-1, a_{23}=13$ ; 5)  $a_{31}=13, a_{23}=0$ .

6. Задан одномерный массив  $x[1..N]$ . Фрагмент алгоритма

```
T:=0; нц для k от 1 до N
| если (k=1) или (T<x[k])
|| то T:=x[k]
| все
кц
```

определяет:

- 1) максимальный элемент массива; 2) сумму элементов, больших T;
- 3) количество элементов, больших T; 4) индекс последнего из элементов, больших T;
- 5) индекс первого из элементов, больших T.

7. Задан двумерный массив  $x[1..n,1..n]$ . В результате работы фрагмента алгоритма

```
нц для i от 1 до div(n,2)
| нц для j от 1 до n
|| c:=a[i,j]; a[i,j]:=a[n-i+1,j]; a[n-i+1,j]:=c;
| кц
кц
```

значения переменных будут соответственно равны

- 1)  $a_{31}=3, a_{12}=5$ ; 2)  $a_{31}=10, a_{12}=6$ ; 3)  $a_{31}=6, a_{12}=3$ ; 4)  $a_{31}=7, a_{12}=9$ ; 5)  $a_{31}=5, a_{12}=2$ .

8. Задан двумерный массив (таблица)  $x[1..n,1..n]$ . Фрагмент алгоритма

```
нц для i от 1 до n
| c:=a[i,i]; a[i,i]:=a[i,n-i+1]; a[i,n-i+1]:=c;
кц
```

- 1) меняет порядок строк таблицы; 2) меняет порядок столбцов;
- 3) меняет местами элементы главной и побочной диагонали;
- 4) меняет местами элементы i-ой строки и j-го столбца; 5) ничего не меняет в таблице.

9. Для массива  $x[1..N]$  алгоритм

```
s:=0; нц для k от 1 до N
| если  $0 > x[k]$ 
| | то  $s:=s+x[k]$ 
| все
кц
```

определяет:

- 1) минимальный элемент массива; 2) сумму отрицательных элементов;
- 3) количество отрицательных элементов;
- 4) индекс последнего отрицательного элемента; 5) максимальный элемент.

10. Для массива  $X[1..n]$  алгоритм

```
P:=0; for k:=n downto 1 do if  $X[k] <> T$  then P:=k;
```

определяет:

- 1) Номер последнего элемента массива, не равного T;
- 2) Количество элементов массива, не равных T;
- 3) Номер первого элемента массива, не равного T; 4) Номер последнего элемента, равного T;
- 5) Количество элементов, равных T; 6) Ни один из ответов 1-5 не верен.

11. Для массива  $x[1..N]$  алгоритм

```
s:=0; нц для k от 1 до N
| если  $T > x[k]$ 
| | то  $s:=k$ 
| все
кц
```

определяет:

- 1) минимальный элемент массива; 2) сумму элементов, меньших T;
- 3) количество элементов, меньших T; 4) индекс последнего из элементов массива, меньших T;
- 5) индекс первого из элементов, меньших T.

12. Для массива  $x[1..N]$  алгоритм

```
S:=0; нц для k от 1 до N
| если  $T > x[k]$ 
| | то  $S:=S+x[k]$ 
| все
кц
```

определяет:

- 1) минимальный элемент массива; 2) сумму элементов массива, меньших T;
- 3) количество элементов, меньших T; 4) индекс последнего из элементов, меньших T;
- 5) индекс первого из элементов, меньших T.

13. Для массива  $x[1..N]$  алгоритм

```
S:=0; нц для k от 1 до N
| если  $T < x[k]$ 
| | то  $S:=S+x[k]$ 
| все
кц
```

определяет:

- 1) максимальный элемент массива; 2) сумму элементов, больших T;

- 3) количество элементов, больших T; 4) индекс последнего из элементов, больших T;  
5) сумму элементов, меньших T.

14. Задан двумерный массив  $x[1..n, 1..n]$ . Фрагмент алгоритма

```
нц для i от 1 до div(n,2)
| нц для j от 1 до n
| | c:=a[i, j]; a[i, j]:=a[n-i+1, j]; a[n-i+1, j]:=c;
| кц
кц
```

- 1) меняет порядок строк таблицы A; 2) меняет порядок столбцов таблицы A;  
3) меняет местами элементы главной и побочной диагонали в таблице A;  
4) меняет местами элементы i-ой строки и i-го столбца таблицы A; 5) ничего не меняет.

### 3. Оценочные средства (оценочные материалы) для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

#### 2 семестр

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.2)

#### Определение количества выполнений цикла в программе

1. Тело цикла в программе

```
V:=нет; r:=45
нц пока не V
| V:=(r=15); r:=mod(r,4)+15
кц
```

выполнится .. столько-то раз.

2. V:=да; x:=27 нц пока V   V:=(x>=3); x:=div(x,8)+1 кц	3. V:=да; x:=64 нц пока V   V:=(x>=4); x:=div(x,6)+2 кц	4. V:=нет; r:=28 нц пока не V   V:=(r=16); x:=mod(r,8)+13 кц
5. V:=нет; r:=22 нц пока не V   V:=(r=10); r:=mod(r,14)+7 кц	6. V:=да; x:=13 нц пока V   V:=(x>=4); x:=div(x,3)+1 кц	7. V:=да; x:=45 нц пока V   V:=(x>=3); x:=div(x,5)+1 кц
8. V:=да; x:=96 нц пока V   V:=(x>=3); x:=div(x,7)+1 кц	9. p:=1; a:=1; нц пока p<16   a:=2*a; p:=p*a; кц	10. p:=1; a:=1; while p<60 do a:=2*a; p:=p*a;

<p>11. <math>r:=2*x</math>;  while <math>x &lt; r - 0.0001</math> do  <math>x:=\text{sqrt}(x)</math>;  При каком <math>x</math> тело цикла выполнится конечное (<math>&gt;0</math>) число раз ?  1) 0.5 2) 0 3) 2 4) <math>-0.5</math></p>	<p>12.  <math>m:=36</math>; <math>n:=56</math>  нц пока <math>m &lt; n</math>    если <math>m &gt; n</math>    то <math>m:=m-n</math>    иначе <math>n:=n-m</math>    все  кц</p>	<p>13.  <math>B:=\text{да}</math>; <math>x:=11</math>  нц пока <math>B</math>    <math>B:=(x &gt; 5)</math>;  <math>x:=\text{div}(x,2)+1</math>  кц</p>
--	---	---

Примечание: среди вариантов ответа может быть "0 раз" и "бесконечно много раз".

### Определение оператора в программе, реализующей алгоритм

Во фрагмент алгоритма, определяющего указанную характеристику массива, нужно вписать логическое выражение по одному из вариантов.

N	фрагмент алгоритма	характеристика массива	логическое выражение
1	$p:=n$ нц для $i$ от 1 до $n$   если _____     то $p:=p-1$   все кц	число элементов массива $V[1..n]$ , больших $x$	1) $i \leq n$ 2) $V[i] \leq x$ 3) $V[i] \geq x$ 4) $V[i] = x$ 5) $i < x$
2	$i:=0$ нц для $i$ от 1 до $N$   если _____     то $i:=i+1$   все кц	число элементов таблицы $A[1..N]$ , равных $x$	1) $A[i] < x$ 2) $A[i] = x$ 3) $A[i] \leq x$ 4) $i \leq N$ 5) $i \leq x$
3	$d:=0$ ; $N:=K$ нц пока $N > 0$   если _____     то $d:=N$   все   $N:=N-1$ кц	индекс первого элемента таблицы $R[1..K]$ , меньшего $x$	1) $R[n] > x$ 2) $R[n] = x$ 3) $R[n] < x$ 4) $N < x$ 5) $N > R[n]$
4	$m:=0$ ; $N:=K$ нц пока $N > 0$   если _____     то $m:=N$   все   $N:=N-1$ кц	индекс первого элемента таблицы $P[1..K]$ , не равного $x$	1) $P[N] = x$ 2) $P[N] > x$ 3) $P[N] < x$ 4) $N > K$ 5) $P[N] < N$
5	$k:=0$ нц для $i$ от 1 до $N$   если _____	число положительных элементов в массиве $A[1..N]$	1) $i \leq N$ 2) $A[i] < A[1]$ 3) $A[i] > 0$

	<pre>    то k:=k+1   все кц </pre>		<pre> 4) A[i] &lt;= 0 5) i &gt; 0 </pre>
6	<pre> k:=0; нц для i от 1 до N   если _____    то k:=k+A[i]   все кц </pre>	сумма положительных элементов в массиве A[1..N]	<pre> 1) i &lt;= N 2) A[i] &lt; A[1] 3) A[i] &gt; 0 4) A[i] &lt;= 0 5) i &gt; 0 </pre>
7	<pre> m:=0; for i:=n downto 1 do if _____ then m:=i; </pre>	индекс первого элемента массива P[1..n], не равного x	<pre> 1) P[i] = x 2) P[i] &gt; x 3) P[i] &lt;&gt; x 4) i &lt;= n 5) Ни один из ответов 1-4 не верен </pre>
8	<pre> 1) p:=0; for i:=1 to 8 do if String[i] = 'R' then p:=i; 2) k:=9; repeat k:=k-1 until String[k] = 'R'; 3) n:=8; for k:=1 to 8 do if String[k] = 'R' then n:=k; </pre>	Какая из приведенных серий команд определяет место последнего вхождения символа 'R' в строку String длины 8 ?	<pre> 1) 1 2) 2 3) 3 4) все три 5) ни один из ответов 1-4 не верен. </pre>
9	<pre> m:=0; нц для k:=1 до N   если _____    то m:=k   все кц </pre>	индекс последнего элемента массива P[1..N], не равного x	<pre> 1) P[k] = x 2) P[k] &lt; x 3) P[k] &lt;&gt; x 4) i &lt;= N 5) P[k] &lt;&gt; m </pre>

### Значения переменных после выполнения фрагмента программы

Необходимо определить значения указанных переменных после выполнения фрагмента алгоритма

	фрагмент алгоритма	переменные	варианты ответа
1	<pre> k:=65; выбор   ПРИ mod( k,12) = 7: d:=k; p:=да   ПРИ mod( k,12) &lt; 5: d:=2; p:=нет   ПРИ mod( k,12) &gt; 9: d:=3; p:=нет   иначе d:=1; p:=да все </pre>	p d	<pre> 1) p=да, d=1 2) p=нет, d=2 3) p=нет, d=3 4) p=да, d=65 5) p=да, d=2 </pre>
2	<pre> k:=47; выбор   ПРИ mod( k,9) = 5: d:=k; p:=да   ПРИ mod( k,9) &lt; 3: d:=2; p:=нет </pre>	p d	<pre> 1) p=да, d=1 2) p=нет, d=2 3) p=нет, d=3 4) p=да, d=47 </pre>

	ПРИ $\text{mod}(k,9) > 7$ : $d:=3$ ; $p:=\text{нет}$   иначе $d:=1$ ; $p:=\text{да}$ все		5) $p=\text{да}$ , $d=3$
3	$k:=7$ ; выбор   ПРИ $\text{mod}(k,12) = 7$ : $d:=k$ ;   ПРИ $\text{mod}(k,12) \leq 5$ : $d:=2*k-1$ ;   ПРИ $\text{mod}(k,12) > 9$ : $d:=\text{mod}(k,12)$ ;   иначе $d:=\text{div}(k,12)$ все	d	1) 0 2) 7/12 3) 5 4) 7 5) 13
4	$k:=7$ ; $y:=\text{mod}(k,12)$ ; выбор   ПРИ $y=7$ : Вывод $k*k$ ;   ПРИ $y=5$ : Вывод $2*k-1$ ;   ПРИ $y=0$ : Вывод $\text{mod}(k,12)$ ;   иначе Вывод $\text{div}(k,12)$ все	Вывод на печать	1) 0 2) 7/12 3) 5 4) 7 5) 49
5	$k:=4$ ; $y:=\text{mod}(k,11)$ ; выбор   ПРИ $y=7$ : Вывод $k*k$ ;   ПРИ $y=4$ : Вывод $2*k-1$ ;   ПРИ $y=0$ : Вывод $\text{mod}(k,12)$ ;   иначе Вывод $\text{div}(k,12)$ все	Вывод на печать	1) 0 2) 4/11 3) 4 4) 7 5) 16
6	$k:=52$ ; выбор   ПРИ $\text{div}(k,12) = 7$ : $d:=k$ ; $p:=\text{да}$   ПРИ $\text{div}(k,12) < 5$ : $d:=2$ ; $p:=\text{нет}$   ПРИ $\text{div}(k,12) > 9$ : $d:=3$ ; $p:=\text{нет}$   иначе $d:=1$ ; $p:=\text{да}$ все	p d	1) $p=\text{да}$ , $d=1$ 2) $p=\text{нет}$ , $d=2$ 3) $p=\text{нет}$ , $d=3$ 4) $p=\text{да}$ , $d=52$ 5) $p=\text{да}$ , $d=2$
7	$c:=0$ ; $a:=27$ ; $b:=4$ ; $d:=a$ нц пока $d \geq b$   $c:=c+1$ ; $d:=d-2*b$ кц	c d	1) $c=3$ , $d=3$ 2) $c=3$ , $d=2$ 3) $c=2$ , $d=3$ 4) $c=2$ , $d=2$ 5) $c=2$ , $d=4$
8	$c:=0$ ; $a:=16$ ; $b:=9$ ; $d:=3*a-5$ нц пока $d \geq b$   $c:=c+1$ ; $d:=d-b-2$ кц	c d	1) $c=4$ , $d=-2$ 2) $c=3$ , $d=10$ 3) $c=4$ , $d=-1$ 4) $c=3$ , $d=-2$ 5) $c=2$ , $d=-2$
9	$c:=11$ ; $a:=24$ ; $b:=14$ ; $d:=2*a-3$ нц пока $d \geq b$   $c:=c-1$ ; $d:=d-b$ кц	c d	1) $c=8$ , $d=3$ 2) $c=6$ , $d=7$ 3) $c=6$ , $d=6$ 4) $c=7$ , $d=7$

			5) c=8, d=13
10	c:=0; a:=23; b:=11; d:=2*a нц пока d>=b   c:=c-1; d:=d-b кц	c d	1) c=4, d=3 2) c=2, d=5 3) c=5, d=3 4) c=4, d=2 5) c=6, d=2
11	Y:=1; i:=0; K:=2; нц для i от 1 до 6   Y:=K*Y кц	Y	1) -24 2) 32 3) -48 4) 64 5) 78
12	z:=1; repeat z:=z+2 until z=10;	z	1) 9 2) 11 3) 10 4) 1 5) Ни один из ответов 1-4 не верен.
13	z:=0; i:=1; while i>1 do begin z:=z +1/i; i:=i-1 end;	z	1) -2 2) 0 3) 1 4) 2 5) Ни один из ответов 1-4 не верен.
14	P:= not P and ('0'='O') or (ord (c) < ord(d)) если c='Q', d='q', P=TRUE.	P	1) 0 2) TRUE 3) FALSE 4) 1 5) -1
15	a:=1; while a<4 do a:=a+1; write('a=',a);	Вывод на печать	1) a=2a=3a=4a=5 2) a=4 3) a=2a=3a=4 4) a=3 5) Ни один из ответов 1-4 не верен.
16	k:=66; выбор   ПРИ div( k,12) = 7: d:=k; p:=да   ПРИ div( k,12) <=5: d:=2; p:=нет   ПРИ div( k,12) > 9: d:=3; p:=нет   иначе d:=1; p:=да все	р и d	1) p=да, d=1 2) p=нет, d=2 3) p=нет, d=3 4) p=да, d=66 5) p=да, d=2
17	a:=3; b:=4 вывод 'x=', 'a*a+b*b', a+b=7;	Вывод на печать	1) x=25 да 2) x=a*a+b*b да 3) x=a*a+b*b a+b=7 4) x=25

			5) $x=25$ $3+4=7$
18	$a:=1;$ нц пока $a<4$   $a:=a+1$   Вывод 'a=',a; кц	вывод на печать	1) $a=2a=3a=4$ 2) $a=2$ 3 4 3) $a=4$ 4) $a=3$ 5) $a=5$
19	нц для $i$ от 1 до 20   Вывод $a+i, i$ кц	последнее значение $i$ на экране	1) 1 2) 20 3) 400 4) 0 5) 200

### Значения переменных в логическом выражении

Определить, при каких значениях переменных (числовой и логической) выражение истинно (или ложно)

	Выражение		Варианты значений переменных a/b/x и P				
1	$((\ln(a)>a)$ или не P) и $(\sqrt{a}<a**4)$	ис- тин- но	2.5 да	0.9 да	2.3 нет	0.1 нет	0.1 да
2	$(\ln(x)<x)$ и не P и $(\sqrt{x}>x*x)$ или $(2*x=x)$	ис- тин- но	2.5 нет	2.5 да	0.5 да	0.5 нет	2 нет
3	P и $(\sin(y)<0)$ или $((x-y)<(x*x-y*y))$ при $y=-1$	лож- но	-2 нет	1.5 нет	-1 да	1 да	3 нет
4	$((\ln(x)/\ln(1/3)>\ln(0.7)/\ln(1/3))$ и $(\sqrt{x}>x*x)$ и не P	ис- тин- но	0.5 да	0.9 да	0.3 нет	0.9 нет	1.5 да
5	$(\arctg(a)>3/4)$ и $(\sin(a)<\cos(b))$ и не P	ис- тин- но	b=0 да	b=0.9 да	b=0 нет	b=1.5 нет	b=1.5 да
6	P или $(\sin(x)>\sin(3))$ и не $\sqrt{x+0.44}<1.2$	лож- но	1 да	4 нет	1 нет	1.4 да	1.4 нет
7	$(\arctg(x)>3/4)$ или $(\sin(x)<\cos(0))$ и не P	ис- тин- но	-3 нет	1 нет	1.7 да	1/3 да	1/3 нет
8	$((y*y-x)>x*x)$ или $(\cos(x)>0)$ и не P при $y=-1$	лож- но	-0.5 да	0.5 да	3 нет	1 нет	3 да

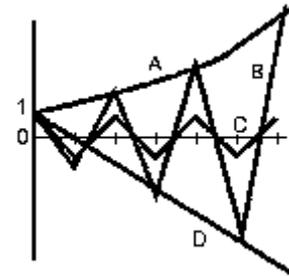
9	$((\ln(x+1) > x) \text{ или не } P)$ и $(\sqrt{x} < x^{**2} + x)$ или $(2*x=x)$	лож- но	0.25 нет	1 нет	3 нет	10 да	0 да
10	$\text{не}(\exp(2*x) > 3.1415/3)$ и не P	ис- тин- но	5 да	0.9 да	4.3 нет	-1 нет	0.9 нет

### Выбор графика по алгоритму

1. Точки с координатами  $(i, Y)$ , последовательно получаемыми при выполнении фрагмента алгоритма (при  $K > 1$ )

```
Y:=1; i:=0
нц для i от 1 до 6
| Y:=-K*Y
кц
```

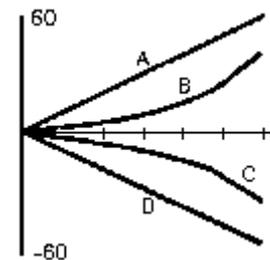
Расположены на графике 1) A; 2) B; 3) C; 4) D; 5)  $Y=0$ .



2. Точки с координатами  $(i, Y)$ , последовательно получаемыми при выполнении фрагмента алгоритма (при  $K > 1$ )

```
Y:=1; i:=0
нц для i от 1 до 5
| Y:=2*Y
кц
```

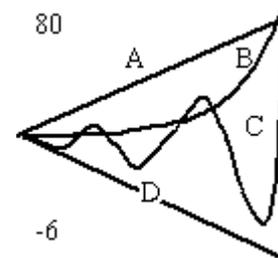
расположены на графике 1) A; 2) B; 3) C; 4) D; 5)  $Y=0$ .



3. Последовательные значения переменной  $Y$  вычисляются по алгоритму

```
Y:=1; i:=0
нц для i от 1 до 6
| Y:=Y*i
кц
```

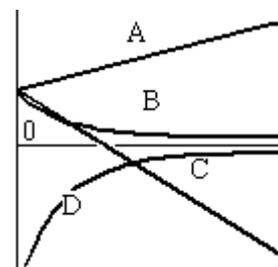
Точки с координатами  $(i, Y)$  расположены на графике 1) A; 2) B; 3) C; 4) D; 5)  $X=0$ .



4. Последовательные значения  $Y$  при  $K > 0$  вычисляются по алгоритму

```
Y:=1; i:=0
нц для i от 1 до 5
| Y:=Y-K
кц
```

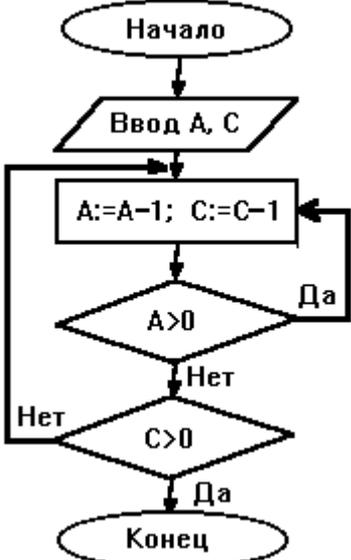
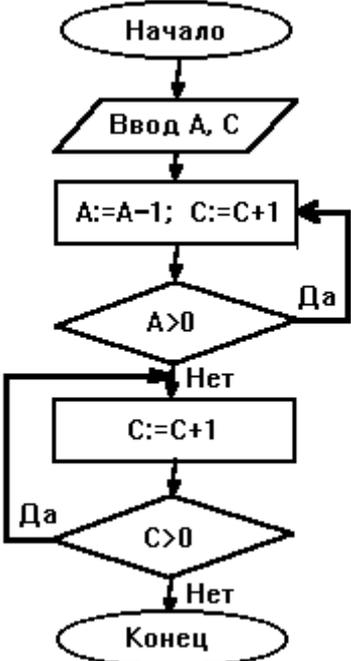
Точки с координатами  $(i, Y)$  расположены на графике 1) A; 2) B; 3) C; 4) D; 5)  $Y=0$ .



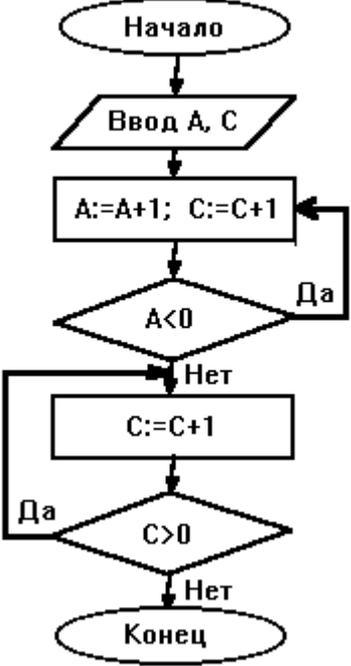
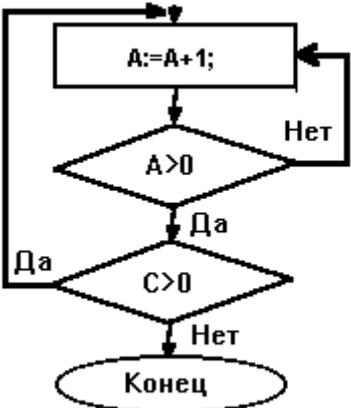
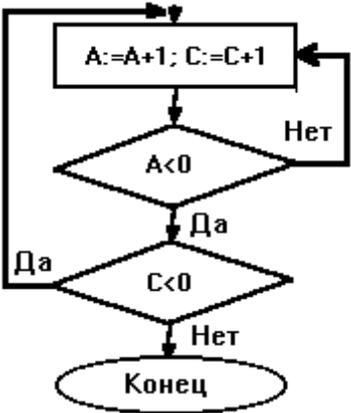
**При каких начальных значениях переменных алгоритм закончит работу**

N	вариант ответа	схема алгоритма
---	----------------	-----------------

1	<p>1) <math>A=-2; C=-3</math>  2) <math>A=-3; C=-2</math>  3) <math>A=-3, C=-3</math>  4) <math>A=-2; C=-1</math>  5) <math>A=-4, C=-3</math></p>	<pre> graph TD     Start([Начало]) --&gt; Input[/Ввод А, С/]     Input --&gt; Process[A:=A+1; C:=C+1]     Process --&gt; DecA{A&gt;0}     DecA -- Нет --&gt; Process     DecA -- Да --&gt; DecC{C&gt;0}     DecC -- Нет --&gt; End([Конец])     DecC -- Да --&gt; Process </pre>
2	<p>1) <math>A=-2; C=-1</math>  2) <math>A=-2; C=-3</math>  3) <math>A=-3, C=3</math>  4) <math>A=-3; C=-2</math>  5) <math>A=-4, C=-3</math></p>	<pre> graph TD     Start([Начало]) --&gt; Input[/Ввод А, С/]     Input --&gt; Process[A:=A+1; C:=C+1]     Process --&gt; DecA{A&lt;0}     DecA -- Нет --&gt; Process     DecA -- Да --&gt; DecC{C&gt;0}     DecC -- Нет --&gt; End([Конец])     DecC -- Да --&gt; Process </pre>
3	<p>1) <math>A=-2; C=1</math>  2) <math>A=-3; C=-4</math>  3) <math>A=-3, C=3</math>  4) <math>A=-3; C=2</math>  5) <math>A=-4, C=3</math></p>	<pre> graph TD     Start([Начало]) --&gt; Input[/Ввод А, С/]     Input --&gt; Process[A:=A+1; C:=C-1]     Process --&gt; DecA{A&lt;0}     DecA -- Нет --&gt; Process     DecA -- Да --&gt; DecC{C&lt;0}     DecC -- Нет --&gt; End([Конец])     DecC -- Да --&gt; Process </pre>

4	<p>1) <math>A=2; C=2</math>  2) <math>A=-3; C=-2</math>  3) <math>A=-3, C=0</math>  4) <math>A=-3; C=-1</math>  5) <math>A=4, C=5</math></p>	 <pre> graph TD     Start([Начало]) --&gt; Input[/Ввод A, C/]     Input --&gt; Process[A:=A-1; C:=C-1]     Process --&gt; Cond1{A&gt;0}     Cond1 -- Да --&gt; Process     Cond1 -- Нет --&gt; Cond2{C&gt;0}     Cond2 -- Да --&gt; End([Конец])     Cond2 -- Нет --&gt; End </pre>
5	<p>1) <math>A=2; C=-3</math>  2) <math>A=3; C=-2</math>  3) <math>A=-3, C=-1</math>  4) <math>A=-3; C=0</math>  5) <math>A=4, C=-4</math></p>	 <pre> graph TD     Start([Начало]) --&gt; Input[/Ввод A, C/]     Input --&gt; Process[A:=A-1; C:=C+1]     Process --&gt; Cond1{A&gt;0}     Cond1 -- Да --&gt; Process     Cond1 -- Нет --&gt; Process2[C:=C+1]     Process2 --&gt; Cond2{C&gt;0}     Cond2 -- Да --&gt; Process2     Cond2 -- Нет --&gt; End([Конец]) </pre>

6	<p>1) A=-2; C=-3  2) A=-3; C=-5  3) A=-3, C=-3  4) A=-3; C=-2  5) A=-4, C=-5</p>	<pre> graph TD     Start([Начало]) --&gt; Input[/Ввод А, С/]     Input --&gt; Process1[A:=A+1; C:=C+1]     Process1 --&gt; Decision1{A&gt;0}     Decision1 -- Да --&gt; Process2[C:=C+1]     Decision1 -- Нет --&gt; End([Конец])     Process2 --&gt; Decision2{C&gt;0}     Decision2 -- Да --&gt; Process1     Decision2 -- Нет --&gt; End </pre>
7	<p>1) A=2; C=2  2) A=-3; C=1  3) A=3, C=4  4) A=-3; C=-2  5) A=4, C=3</p>	<pre> graph TD     Start([Начало]) --&gt; Input[/Ввод А, С/]     Input --&gt; Process1[A:=A-1; C:=C-1]     Process1 --&gt; Decision1{A&lt;0}     Decision1 -- Да --&gt; Process1     Decision1 -- Нет --&gt; Process2[C:=C-1]     Process2 --&gt; Decision2{C&lt;0}     Decision2 -- Да --&gt; Process1     Decision2 -- Нет --&gt; End([Конец]) </pre>
8	<p>1) A=-2; C=-2  2) A=-3; C=-2  3) A=-3, C=3  4) A=-3; C=-3  5) A=-4, C=-5</p>	<pre> graph TD     Start([Начало]) --&gt; Process1[A:=A-1; C:=C+1]     Process1 --&gt; Decision1{A&lt;0}     Decision1 -- Да --&gt; Decision2{C&lt;0}     Decision1 -- Нет --&gt; Process1     Decision2 -- Да --&gt; End([Конец])     Decision2 -- Нет --&gt; Process1 </pre>

9	<p>1) <math>A=-2; C=-3</math>  2) <math>A=-3; C=-2</math>  3) <math>A=-3, C=-3</math>  4) при любых <math>A</math> и <math>C</math>  5) <math>A=-4, C=-3</math></p>	 <pre> graph TD     Start([Начало]) --&gt; Input[/Ввод А, С/]     Input --&gt; Process1[A:=A+1; C:=C+1]     Process1 --&gt; Decision1{A&lt;0}     Decision1 -- Да --&gt; Process1     Decision1 -- Нет --&gt; Process2[C:=C+1]     Process2 --&gt; Decision2{C&gt;0}     Decision2 -- Да --&gt; Process2     Decision2 -- Нет --&gt; End([Конец]) </pre>
10	<p>1) <math>A=-3; C=-2</math>  2) <math>A=-2; C=-3</math>  3) <math>A=-3, C=-3</math>  4) при любых <math>A</math> и <math>C</math>  5) <math>A=-1, C=2</math></p>	 <pre> graph TD     Start([Начало]) --&gt; Process1[A:=A+1]     Process1 --&gt; Decision1{A&gt;0}     Decision1 -- Нет --&gt; Process1     Decision1 -- Да --&gt; Decision2{C&gt;0}     Decision2 -- Да --&gt; Process1     Decision2 -- Нет --&gt; End([Конец]) </pre>
11	<p>1) <math>A=2; C=-4</math>  2) <math>A=1; C=-2</math>  3) <math>A=3, C=-4</math>  4) при любых <math>A</math> и <math>C</math>  5) при любом <math>C</math> и <math>A=2</math></p>	 <pre> graph TD     Start([Начало]) --&gt; Process1[A:=A+1; C:=C+1]     Process1 --&gt; Decision1{A&lt;0}     Decision1 -- Нет --&gt; Process1     Decision1 -- Да --&gt; Decision2{C&lt;0}     Decision2 -- Да --&gt; Process1     Decision2 -- Нет --&gt; End([Конец]) </pre>

## Определение результата работы программы

1. Определить значение переменной  $k$  после выполнения фрагмента программы.

N	фрагмент программы	N	фрагмент программы
1. 1	$k:=0;$ нц для $i:=1$ до 1000   если $\text{mod}(i, 3)=1$ и $\text{mod}(i, 5)=2$     то $k:=k+1$   все кц	1. 2	$k:=0;$ нц для $i:=1$ до 1000   если $\text{mod}(i, 3)=1$ или $\text{div}(i, 4)=200$     то $k:=k+1$ кц
1. 3	$k:=1$ нц для $i:=1$ до 1000   если $\text{mod}(i, 4)=1$ и не $\text{mod}(i, 5)=2$     то $k:=k+1$   все кц	1. 4	$k:=0$ нц для $i:=1$ до 1000   если $\text{mod}(i, 3)=1$ или $\text{mod}(i, 3)=2$     то $k:=k+1$   все кц
1. 5	$k:=0$ нц для $i:=1$ до 1000   если не $\text{div}(i, 3)=100$ и $\text{mod}(i, 5)=2$     то $k:=k+1$   все кц	1. 6	$k:=1$ нц для $i:=1$ до 1000   если не $\text{div}(i, 300)=3$ и $\text{mod}(i, 3)=1$     то $k:=k+1$   все кц
1. 7	$k:=0$ нц для $i:=1$ до 1000   если не $\text{div}(i, 100)=3$ и $\text{mod}(i, 5)=2$     то $k:=k+1$   все кц	1. 8	$k:=0$ нц для $i:=1$ до 1000   если $\text{mod}(i, 4)=1$ или $\text{mod}(i, 5)=2$     то $k:=k+1$   все кц
1. 9	$k:=1;$ нц для $i:=1$ до 1000   если $\text{mod}(i, 3)=1$ или $\text{mod}(i, 4)=3$     то $k:=k+1$   все кц		

2. После выполнения фрагмента алгоритма переменная  $k$  приняла указанное значение. Определить минимальное целое  $N$ , при котором это возможно.

	фрагмент алгоритма	$k$	$N=?$
2.1	$k:=0; i:=1;$ нц пока $i \leq N$   если $\text{mod}(i, 3)=0$     то $k:=k+1$   все	16	12 13 14 15 16

	i:=i+1 кц k:=k*k-2*k+1		
2.2	k:=0; i:=1; нц пока i<=N   i:=i+1   если mod(i, 2)=0 и mod(i, 4)<>0     то k:=k+1   все кц k:=(k+1)*(k*k-k+1)	217	20 21 22 23 24
2.3	k:=0; i:=1; нц пока k<=N   i:=i+1   если mod(i, 2)=0 и mod(i, 4)<>0     то k:=k+1   все кц k:=k*k-2*k+1	16	6 4 8 9 7

#### 4. Оценочные средства (оценочные материалы) для проведения промежуточной аттестации обучающихся (защиты курсовой работы (проекта)) по дисциплине (модулю)

##### 2 семестр

Задание: подготовить программным модуль на заданную тематику.

##### Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.3)

Проверяется:

- актуальность использованной версии программного обеспечения
- использование сетевых источников информации
- удобство использования разработанного программного модуля с точки зрения пользователя

##### Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-6 (контролируемый индикатор достижения компетенции ОПК-6.1)

Проверяется:

- соответствие выполненной работы заданию
- алгоритм решения
- использованные в программном коде типовые модули
- работоспособность разработанного программного модуля при различных исходных данных
- блокировка программой явно неверного ввода данных