

Негосударственное частное образовательное учреждение
высшего образования
"Алтайский экономико-юридический институт"
Кафедра государственно-правовых дисциплин

УТВЕРЖДАЮ
Ректор Алтайского экономико-
юридического института
В. И. С. Иванов
" 24 " ноября 2016 г.



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине

Вычислительная техника и основы программирования

для направления 38.03.01 Экономика
квалификация (степень) "бакалавр"
Профиль подготовки
"Финансы и кредит"

Барнаул 2016

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Область применения

Фонд оценочных средств – является неотъемлемой частью учебно-методического комплекса учебной дисциплины «Вычислительная техника и основы программирования» и предназначен для контроля и оценки образовательных достижений обучающихся, освоивших программу данной дисциплины.

1.2. Контролируемые компетенции

Код контролируемой компетенции	Этап формирования компетенции	Способ оценивания	Оценочное средство
ОПК-1: способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	базовый	Зачет	Комплект контролирующих материалов для зачета
ПК-8: способность использовать для решения аналитических и исследовательских задач современные технические средства и информационные технологии	базовый	Зачет	Комплект контролирующих материалов для зачета
ПК-10: способностью использовать для решения коммуникативных задач современные технические средства	базовый	Зачет	Комплект контролирующих материалов для зачета

и информационные технологии			
-----------------------------	--	--	--

Показатели оценивания компетенций представлены в разделе «Требования к результатам освоения дисциплины» рабочей программы дисциплины «Вычислительная техника и основы программирования» с декомпозицией: знать, уметь, владеть.

При оценивании сформированности компетенций по дисциплине «Вычислительная техника и основы программирования» используется 100-балльная шкала.

Профессиональный уровень “5” (отлично)	85-100	<p>Ответ хорошо структурирован; полное понимание исследуемого вопроса; полный и глубокий анализ вопроса; критическое использование теории и рекомендуемого материала для чтения; расширение и углубление лекционного материала; аргументированная логика; продуманность, творческий и оригинальный подход к освещению вопроса; иллюстративность массой примеров и данных</p>
Продвинутый уровень “4” (хорошо)	70-84	<p>Хорошая организация, но ряд несущественных упущений в плане содержания; умение аргументировать и использовать примеры; некоторое расширение и углубление лекционного материала; использование соответствующих концептуальных моделей</p>
Базовый уровень “3” (удовлетворительно)	60-69	<p>Удовлетворительный уровень, есть ряд существенных упущений; слабые места в стилевом оформлении, структуре и анализе; в основном базируется на лекционном материале; информация представлена четко, но отсутствует оригинальность в ее изложении</p>
Минимальный уровень “2” (неудовлетворительно)	35-59	<p>Неудовлетворительное выполнение; частичное понимание проблемы; несмотря на наличие ряда весьма удачных мест, работа характеризуется отсутствием тщательного анализа; неадекватность примеров</p>

Минимальный уровень “1” (неудовлетворительно)	0-34	Отсутствие понимания вопроса, работа не структурирована и не соответствует требованиям; наличие серьезных ошибок и несоответствий
---	------	---

Рейтинговая система для оценки успеваемости студентов

Разбивка баллов.

Промежуточный рейтинг – 70 баллов:

1) Рейтинг работы студента на практических занятиях – 22 балла.

Максимальный рейтинг, который студент может заработать на одном семинарском занятии – 2 балла:

- за отличный ответ (полный, безошибочный) – 2 балла;
- за активную работу на семинаре (от 2 до 4 выступлений) – 1-2 балла;
- за неточное выступление, за неточное дополнение — 1 балл;
- за отказ от ответа, за неправильный ответ – 0 баллов.

2) Рейтинг контрольных точек – 25 баллов.

3) Рейтинг посещения лекционных занятий – 6 баллов.

4) Рейтинг посещения семинарских занятий – 7 баллов.

5) Рейтинг поощрительный – 10 баллов:

- разработка сценария деловой игры – 10 баллов;
- составление кроссвордов – 5 баллов;
- решение задач повышенной сложности – 5-10 баллов;
- Написание и защита реферата – 3-7 баллов.

Сдача зачета – 30 баллов.

Пересчет суммы баллов в традиционную и международную оценку

Оценка (ФГОС)	Итоговая сумма баллов, учитывает успешно сданный экзамен	Оценка (ECTS)
5 (отлично) зачтено	90 - 100	A (отлично)
4 (хорошо) зачтено	85 – 89	B (очень хорошо)
	75 – 84	C (хорошо)
	70 - 74	D (удовлетворительно)
65 – 69		
3 (удовлетворительно) зачтено	60 - 64	E (посредственно)
	Ниже 60 баллов	F (неудовлетворительно)

2. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ

2.1. Оценочные средства для текущего контроля успеваемости

Задачи:

Задача 1.

1. Решить задачу: Крестьянину нужно перевезти через реку волка, козу и капусту. Но лодка такова, что в ней может поместиться только крестьянин, а с ним или один волк, или одна коза, или одна капуста. Но если оставить волка с козой, то волк съест козу, а если оставить козу с капустой, то коза съест капусту. Как перевез свой груз крестьянин? (Задача, придуманная ученым монахом и математиком из Ирландии Алкуином (735–804))
2. Записать словесный алгоритм решения задачи.
3. Записать решение задачи в виде блок-схемы.

Задача 2. Разработать программу на языке Си. Необходимо вывести текст в соответствии с вариантом задания.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. Составить тестовые примеры с ответами.
4. Протестировать программу на подготовленных примерах.
5. Составить отчёт о проделанной работе.
6. Защитить работу.

Задача 3. Разработать программу на языке Си реализующую линейный алгоритм. Необходимо решить математическую функцию в соответствии с вариантом задания.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.

3. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
4. Протестировать программу на подготовленных примерах.
5. Составить отчёт о проделанной работе.
6. Защитить работу.

Задача 4. Разработать программу на языке Си реализующую алгоритм ветвления. Необходимо решить математическую функцию в соответствии с вариантом задания для решения использовать оператор `if`.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
4. Протестировать программу на подготовленных примерах.
5. Составить отчёт о проделанной работе.
6. Защитить работу.

Задача 5. Разработать программу на языке Си реализующую алгоритм ветвления. Необходимо решить математическую функцию в соответствии с вариантом задания для решения использовать оператор `switch`.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
4. Протестировать программу на подготовленных примерах.
5. Составить отчёт о проделанной работе.
6. Защитить работу.

Задача 6. Разработать программу на языке Си реализующую циклический алгоритм. Необходимо решить математическую функцию в соответствии с вариантом задания для решения использовать оператор `while`.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска решения математических функций с помощью оператора `do`.
4. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Задача 7. Разработать программу на языке Си реализующую циклический алгоритм. Необходимо решить математическую функцию в соответствии с вариантом задания для решения использовать оператор `for`.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска решения математических функций с помощью оператора `do`.
4. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Задача 8. Разработать программу на языке Си реализующую циклический алгоритм. Необходимо решить математическую функцию в соответствии с вариантом задания для решения использовать оператор do.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска решения математических функций с помощью оператора do.
4. Составить тестовые примеры и подготовить для них ответы (ответы рассчитать с помощью калькулятора).
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Задача 9. Разработать программу на языке Си, реализующую работу с одномерным массивом. Необходимо найти минимальный и максимальный элемент массива, посчитать количество положительных и отрицательных элементов (0 не является положительным или отрицательным числом), а так же выполнить дополнительное задание согласно элементу.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска с помощью оператора do.
4. Составить тестовые примеры и подготовить для них ответы.
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Задача 10. Разработать программу на языке Си, реализующую работу с двумерным массивом, состоящим из чисел. Необходимо найти минимальный и максимальный элемент массива, посчитать количество положительных и отрицательных элементов (0 не является положительным или отрицательным числом), найти сумму всех элементов на главной диагонали, а так же выполнить дополнительное задание согласно элементу.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска с помощью оператора do.
4. Составить тестовые примеры и подготовить для них ответы.
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Задача 11. Разработать программу на языке Си, реализующую работу с двумерным массивом, состоящим из чисел. Все основные переменные будут динамическими. Необходимо найти минимальный и максимальный элемент массива, посчитать количество положительных и отрицательных элементов (0 не является положительным или отрицательным числом), найти сумму всех элементов на главной диагонали, а так же выполнить дополнительное задание согласно элементу.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска с помощью оператора do.
4. Составить тестовые примеры и подготовить для них ответы.
5. Протестировать программу на подготовленных примерах.

6. Составить отчёт о проделанной работе.
7. Защитить работу.

Задача 12. Разработать программу на языке Си, реализующую работу с двумерным массивом, состоящим из чисел. Необходимо найти минимальный и максимальный элемент массива, посчитать количество положительных и отрицательных элементов (0 не является положительным или отрицательным числом), найти сумму всех элементов на главной диагонали, а так же выполнить дополнительное задание согласно элементу. Все математические алгоритмы необходимо оформить функциями.

Для выполнения задачи необходимо:

1. Составить блок-схему.
2. Написать программу.
3. В программе предусмотреть возможность повторного запуска с помощью оператора do.
4. Составить тестовые примеры и подготовить для них ответы.
5. Протестировать программу на подготовленных примерах.
6. Составить отчёт о проделанной работе.
7. Защитить работу.

Тестовые задания:

1. Между населёнными пунктами А, В, С, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	A	B	C	D	E	F
A			3			
B			9		4	
C	3	9		3	8	
D			3		2	
E		4	8	2		7

F					7	
---	--	--	--	--	---	--

Определите длину кратчайшего пути между пунктами А и F (при условии, что передвигаться можно только по построенным дорогам).

- 1-11
- 2-13
- 3-15**
- 4-17

2. Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность. Вот этот код: А–10, Б–001, В–0001, Г–110, Д–111.

Можно ли сократить для одной из букв длину кодового слова так, чтобы код по-прежнему можно было декодировать однозначно? Коды остальных букв меняться не должны. Выберите правильный вариант ответа.

- 1 это невозможно
- 2 для буквы В — 000**
- 3 для буквы Б — 0
- 4 для буквы Г — 11

3. В программе описан одномерный целочисленный массив с индексами от 0 до n. Ниже представлен фрагмент одной и той же программы, записанный на разных языках программирования, обрабатывающей данный массив:

```

s = 0;
z = A[n];
for (i = 0; i <= n; i++){
  if (A[i] > z)
    s++;

```

}

Чему будет равно значение переменной s после выполнения данной программы, при любых значениях элементов массива?

1 Максимальному элементу в массиве A

2 Количеству элементов массива A , больших последнего элемента массива

3 Индексу последнего элемента массива A , который меньше $A[0]$

4 Количеству элементов массива A , меньших последнего элемента массива

4. В программе описан одномерный целочисленный массив с индексами от 0 до n (т.е. первый элемент имеет индекс 0, последний - индекс n). Ниже представлен фрагмент одной и той же программы, записанный на разных языках программирования, обрабатывающей данный массив:

Си
<pre>s = n; z = A[0]; for (i = 1; i <= n; i++){ if (A[i] == z) s--; }</pre>

Чему будет равно значение переменной s после выполнения данной программы, при любых значениях элементов массива?

1 Количеству элементов массива A , больших первого элемента массива

2 Количеству элементов массива A , не превосходящих первого элемента массива

3 Количеству элементов массива A , не равных первому элементу массива

4 Количеству элементов массива A , равных первому элементу массива

5. В программе описан одномерный целочисленный массив с индексами от 0 до n . Ниже представлен записанный на разных языках

программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Си
<pre>s = 0; z = A[n]; for (i = 0; i < n; i++) if (A[i] < z) s=s + A[i];</pre>

Чему будет равно значение переменной *s* после выполнения данной программы? Ответ должен быть верным при любых значениях элементов массива.

- 1 Минимальному элементу в массиве *A*
- 2 Количеству элементов массива *A*, меньших последнего элемента массива
- 3 Сумме всех элементов массива *A*, меньших последнего элемента массива**
- 4 Индексу первого элемента массива *A*, который меньше *A[n]*

6. В программе описан одномерный целочисленный массив с индексами от 0 до *n*. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Си
<pre>s = 0; z = A[n]; for (i = 0; i <= n-1; i++) if (A[i] > z) s=s+ A[i];</pre>

Чему будет равно значение переменной *s* после выполнения данной программы? Ответ должен быть верным при любых значениях элементов массива.

- 1 Максимальному элементу в массиве *A*

2Количество элементов массива A, больших последнего элемента массива

3Сумме всех элементов массива A, больших последнего элемента массива

4Индексу последнего элемента массива A, который больше A[n]

7. В программе описан одномерный целочисленный массив с индексами от 0 до n (т.е. первый элемент имеет индекс 0, последний - индекс n). Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Си
<pre>s = 0; for (i = 1; i <= n; i++) if (A[i] == A[0]) s=i;</pre>

Чему будет равно значение переменной s после выполнения данной программы? Ответ должен быть верным при любых значениях элементов массива.

1Минимальному элементу в массиве A

2Количеству элементов массива A, равных первому элемента массива

3Сумме всех элементов массива A, равных последнему элементу массива

4Наибольшему индексу k, для которого элемент массива с индексом k равен первому элементу массива

8. В программе описан одномерный целочисленный массив с индексами от 0 до 10. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Бейсик	Паскаль
<pre>s = 0 n = 10 FOR i = 1 TO n IF A(i)-A(i-1) > 10 THEN s = s+A(i) END IF NEXT i</pre>	<pre>s := 0; n := 10; for i := 1 to n do begin if A[i]-A[i-1] > 10 then s := s+A[i]; end;</pre>
Си	Алгоритмический язык
<pre>s = 0; n = 10; for (i = 1; i <= n; i++) if(A[i]-A[i-1] > 10) s = s+A[i];</pre>	<pre>s := 0 n := 10 нц для i от 1 до n если A[i]-A[i-1] > 10 то s := s+A[i] все кц</pre>

В начале выполнения этого фрагмента в массиве находились числа 0,10,20,30,40,50,60,70,80,90,100, т.е. $A[0]=0$, $A[1]=10$ и т.д. Чему будет равно значение переменной s после выполнения данной программы?

- 1 0
- 2 10
- 3 30
- 4 550

9. В программе описан одномерный целочисленный массив с индексами от 0 до 10. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Бейсик	Паскаль
<pre>s = 0 n = 10 FOR i = 1 TO n IF A(i)-A(i-1) < i THEN s = s+i END IF NEXT i</pre>	<pre>s := 0; n := 10; for i := 1 to n do begin if A[i]-A[i-1] < i then s := s+i; end;</pre>
Си	Алгоритмический язык
<pre>s = 0; n = 10; for (i = 1; i <= n; i++) if(A[i]-A[i-1] < i) s = s+i;</pre>	<pre>s := 0 n := 10 нц для i от 1 до n если A[i]-A[i-1] < i то s := s+i все кц</pre>

В начале выполнения этого фрагмента в массиве находились числа Фибоначчи: 1,1,2,3,5,8,13,21,34,55,89, т.е. $A[0]=1$, $A[1]=1$, $A[2]=2$ и т.д. Чему будет равно значение переменной s после выполнения данной программы?

Условие $A[i]-A[i-1]<i$ выполнено $i=1,\dots,6$. Поэтому итоговое значение s равно $1+2+3+4+5+6=(1+6)\cdot 6/2=7\cdot 3=21$.

1 0

2 21

3 34

4 55

10. В программе описан одномерный целочисленный массив с индексами от 0 до 100. Ниже представлен записанный на разных языках программирования фрагмент одной и той же программы, обрабатывающей данный массив:

Чему будет равно значение переменной s после выполнения данной программы?

Бейсик	Паскаль
<pre>N = 100 FOR i = 0 TO n A(i) = i NEXT i FOR i = 1 TO n A(i) = A(i) - A(i-1) NEXT i s = A(90)</pre>	<pre>n := 100; for i := 0 to n do begin A[i] := i; end; for i := 1 to n do begin A[i] := A[i] - A[i-1]; end; s := A[90]</pre>
Си	Алгоритмический язык
<pre>n = 100; for (i = 0; i <= n; i++) A[i] = i for (i = 1; i <= n; i++) A[i] = A[i] - A[i-1]; s = A[90]</pre>	<pre>n := 100 нц для i от 0 до n A[i] := i кц нц для i от 1 до n A[i] := A[i] - A[i-1] кц s := A[90]</pre>

1 0

2 30

3 45

4 550

11. В программе описан одномерный целочисленный массив с индексами от 0 до n. Ниже представлен фрагмент одной и той же программы, записанный на разных языках программирования, обрабатывающей данный массив:

Бейсик	Паскаль
<pre>s = 0 z = A(n) FOR i = 0 TO n IF A(i) > z THEN s = s + 1 NEXT i</pre>	<pre>s:=0; z:=A[n]; for i:=0 to n do begin if A[i]>z then s:=s+1; end</pre>
Си	Алгоритмический язык
<pre>s = 0;</pre>	<pre>s:=0</pre>

<pre> z = A[n]; for (i = 0; i <= n; i++){ if (A[i] > z) s++; } </pre>	<pre> z:=A[n] нцдля i от 0 до n если A[i]> z то s:=s+1 все кц </pre>
---	---

Чему будет равно значение переменной s после выполнения данной программы, при любых значениях элементов массива?

1 Максимальному элементу в массиве A

2 **Количеству элементов массива A, больших последнего элемента массива**

3 Индексу последнего элемента массива A, который меньше A[0]

4 **Количеству элементов массива A, меньших последнего элемента массива**

12. Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо
-------	------	-------	--------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно
-----------------	----------------	----------------	-----------------

Цикл

ПОКА < условие >

последовательность команд

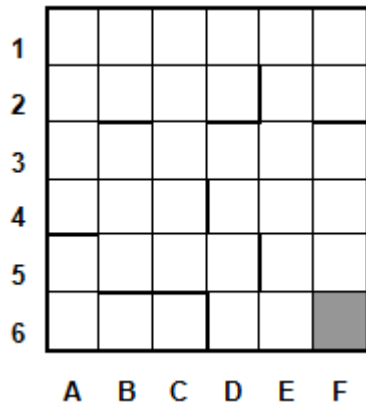
КОНЕЦ ПОКА

выполняется, пока условие истинно.

Если РОБОТ начнёт движение в сторону находящейся рядом с ним стены, то

он разрушится и программа прервётся.

Схема лабиринта:



Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО

ПОКА<справа свободно ИЛИ снизу свободно>

ПОКА <снизу свободно>

вниз

КОНЕЦ ПОКА

ПОКА <справа свободно>

вправо

КОНЕЦ ПОКА

КОНЕЦ ПОКА

КОНЕЦ

1 18

2 22

3 26

4 30

13. Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо
--------------	-------------	--------------	---------------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно
------------------------	-----------------------	-----------------------	------------------------

Цикл

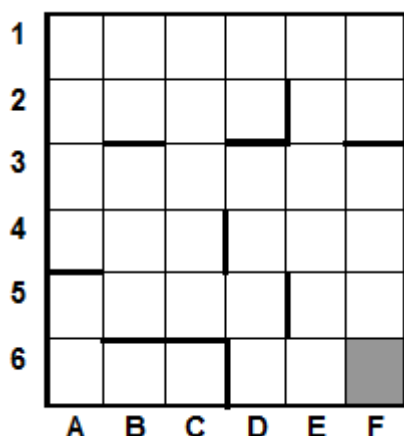
ПОКА < условие >

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

Если РОБОТ начнёт движение в сторону находящейся рядом с ним стены, то он разрушится и программа прервётся. Схема лабиринта:



Сколько клеток лабиринта соответствуют требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО

ПОКА<справа свободно ИЛИ снизу свободно>

ПОКА <справа свободно>

вправо

КОНЕЦ ПОКА

ПОКА <снизу свободно>

вниз

КОНЕЦ ПОКА

КОНЕЦ ПОКА

КОНЕЦ

1 18

2 21

3 24

4 27

14. У исполнителя Калькулятор две команды, которым присвоены номера:

1. **отними 2**
2. **раздели на 3**

Выполняя первую из них, Калькулятор отнимает от числа на экране 2, а выполняя вторую, делит его на 3 (если деление нацело невозможно, Калькулятор отключается).

Запишите порядок команд в программе получения из числа 37 число 3, содержащей не более 5 команд, указывая лишь номера команд.

(Например, программа **2121** – это программа
раздели на 3
отними 2
раздели на 3
отними 2

Эта программа, например, преобразует число 60 в число 4.)

Ответ: 11212

15. Определите значение переменной **c** после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования):

Бейсик	Паскаль
<pre> a = 30 b = 6 a = a / 5 * b IF a > b THEN c = a - 4 * b ELSE c = a + 4 * b ENDIF </pre>	<pre> a := 30; b := 6; a := a / 5 * b; if a > b then c := a - 4 * b else c := a + 4 * b; </pre>
Си	Алгоритмический
<pre> a = 30; b = 6; a = a / 5 * b; if (a > b) c = a - 4 * b; else c = a + 4 * b; </pre>	<pre> a := 30 b := 6 a := a / 5 * b <u>если</u> a > b <u>то</u> c := a - 4 * b <u>иначе</u> c := a + 4 * b <u>все</u> </pre>

Ответ: 12

16. Определите значение переменной **c** после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования):

Бейсик	Паскаль
<pre> a = 30 </pre>	<pre> a := 30; </pre>

<pre> b = 6 a = a / 3 * b IF a > b THEN c = a - 5 * b ELSE c = a + 5 * b ENDIF </pre>	<pre> b := 6; a := a / 3 * b; if a > b then c := a - 5 * b else c := a + 5 * b; </pre>
Си	Алгоритмический
<pre> a = 30; b = 6; a = a / 3 * b; if (a > b) c = a - 5 * b; else c = a + 5 * b; </pre>	<pre> a := 30 b := 6 a := a / 3 * b <u>если</u> a > b <u>то</u> c := a - 5 * b <u>иначе</u> c := a + 5 * b <u>все</u> </pre>

Ответ: 30

17. Определите значение переменной c после выполнения следующего фрагмента программы (записанного ниже на разных языках

программирования):

Бейсик	Паскаль
<pre>a = 60 b = 20 a = a / b * 2 IF a > b THEN c = a - 4 * b ELSE c = a + 4 * b ENDIF</pre>	<pre>a := 60; b := 20; a := a / b * 2; if a > b then c := a - 4 * b else c := a + 4 * b;</pre>
Си	Алгоритмический
<pre>a = 60; b = 20; a = a / b * 2; if (a > b) c = a - 4 * b; else c = a + 4 * b;</pre>	<pre>a := 60 b := 20 a := a / b * 2 <u>если</u> a > b <u>тогда</u> := a - 4 * b <u>иначе</u> := a + 4 * b <u>все</u></pre>

Ответ: 86

18. Определите значение переменной с после выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> a = 30 b = 10 a = a - b * 2 IF a > b THEN c = a * 4 * b ELSE c = a * 4 / b ENDIF </pre>	<pre> a := 30; b := 10; a := a - b * 2; if a > b then c := a * 4 * b else c := a * 4 / b; </pre>
Си	Алгоритмический
<pre> a = 30; b = 10; a = a - b * 2; if (a > b) c = a * 4 * b; else c = a * 4 / b; </pre>	<pre> a := 30 b := 10 a := a - b * 2 <u>если</u> a > b <u>то</u> := a*4 * b <u>иначе</u> := a*4 /b <u>все</u> </pre>

Ответ: 4

19. Определите значение переменной с после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования):

Бейсик	Паскаль
<pre> a = 50 b = 10 a = a / b * 2 IF a > b THEN c = a - 4 * b ELSE c = a + 4 * b ENDIF </pre>	<pre> a := 50; b := 10; a := a / b * 2; if a > b then c := a - 4 * b else c := a + 4 * b; </pre>
Си	Алгоритмический
<pre> a = 50; b = 10; a = a / b * 2; if (a > b) c = a - 4 * b; else c = a + 4 * b; </pre>	<pre> a := 50 b := 10 a := a / b * 2 если a > b <u>то</u> := a - 4 * b <u>иначе</u> := a + 4 * b <u>все</u> </pre>

Правильно: **50**

20. **В6.6.** Определите значение переменной **c** после выполнения следующего фрагмента программы (*записанного ниже на разных языках программирования*):

Бейсик	Паскаль
<pre> a = 50 b = 10 a = a / b * 2 IF a >= b THEN c = (a - b) * (a + b) ELSE </pre>	<pre> a := 50; b := 10; a := a / b * 2; if a >= b then c := (a - b) * (a + b); else </pre>

$c = a + 4 * b$ <i>ENDIF</i>	$c := a + 4 * b;$
Си	Алгоритмический
$a = 50;$ $b = 10;$ $a = a / b * 2;$ <i>if</i> ($a >= b$) $c = (a - b) * (a + b);$ <i>else</i> $c = a + 4 * b;$	$a := 50$ $b := 10$ $a := a / b * 2$ <i>если</i> $a >= b$ <i>то</i> $c := (a - b) * (a + b)$ <i>иначе</i> $c := a + 4 * b$ <i>все</i>

Решение: После присваивания $a = a / b * 2$, значение a станет равным $50 / 10 * 2 = (50/10) * 2 = 5 * 2 = 10$.

Значение выражения $a >= b$ – истина, поэтому в условном операторе будет выполнен вариант THEN (то) т.е. присваивание $c = (a - b) * (a + b)$. При этом значение c станет равным $(10 - 10) * (10 + 10) = 0 * 20 = 0$.

Ответ: 0

21. Определите значение переменной c после выполнения следующего фрагмента программы (записанного ниже на разных языках

программирования). Ответ запишите с точностью до целого.

Бейсик	Паскаль
<pre>a = 30 b = 6 a = a/5 - b IF a > b THEN c = a - 5*b ELSE c = a + 5*b ENDIF</pre>	<pre>a := 30; b := 6; a := a/5 - b; if a > b then c := a - 5*b else c := a + 5*b;</pre>
Си	Алгоритмический
<pre>a = 30; b = 6; a = a/5 - b; if (a > b) c = a - 5*b; else c = a + 5*b;</pre>	<pre>a := 30 b := 6 a := a/5 - b если a > b то c := a - 5*b иначе c := a + 5*b все</pre>

Ответ: 30

22. Определите значение переменной c после выполнения следующего фрагмента программы (записанного ниже на разных языках

программирования). Ответ запишите с точностью до целого.

Бейсик	Паскаль
<pre>a = 30 b = 6 a = a/5 - b/3 IF 2*a > b THEN c = a - 5*b + 27 ELSE c = a + 5*b + 27 ENDIF</pre>	<pre>a := 30; b := 6; a := a/5 - b/3; if 2*a > b then c := a - 5*b + 27 else c := a + 5*b + 27;</pre>
Си	Алгоритмический
<pre>a = 30; b = 6; a = a/5 - b/3; if (2*a > b) c = a - 5*b + 27; else c = a + 5*b + 27;</pre>	<pre>a := 30 b := 6 a := a/5 - b/3 если 2*a > b то c := a - 5*b + 27 иначе c := a + 5*b + 27 все</pre>

Ответ: 1

23. Определите значение переменной **c** после выполнения следующего фрагмента программы (записанного ниже на разных языках программирования). Ответ запишите с точностью до целого.

Бейсик	Паскаль
<pre>a = 30 b = 6 a = a/5 - (b - 3) IF 2*a > b THEN c = a - 5*b + 2 ELSE c = a + 5*b + 2 ENDIF</pre>	<pre>a := 30; b := 6; a := a/5 - (b - 3); if 2*a > b then c := a - 5*b + 2 else c := a + 5*b + 2;</pre>
Си	Алгоритмический
<pre>a = 30; b = 6; a = a/5 - (b - 3); if (2*a > b) c = a - 5*b + 2; else c = a + 5*b + 2;</pre>	<pre>a := 30 b := 6 a := a/5 - (b - 3) если 2*a > b то c := a - 5*b + 2 иначе c := a + 5*b + 2 все</pre>

Ответ: 35

24. Определите, что будет напечатано в результате выполнения программы, записанной ниже на разных языках программирования:

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 1 S = 0 WHILE N <= 101 S = S + 7 N = N + 1 WEND PRINT S </pre>	<pre> var n, s: integer; begin n := 1; s := 0; while n <= 101 do begin s := s + 7; n := n + 1 end; write(s) end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main() { int n, s; n = 1; s = 0; while (n <= 101) { s = s + 7; n = n + 1; } printf("%d", s); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> n, s n := 1 s := 0 <u>нцпока</u> n <= 101 s := s + 7 n := n + 1 <u>кц</u> <u>вывод</u> s <u>кон</u> </pre>

Ответ: 707

25. (ege.yandex.ru-2)

Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre><i>DIM N, S AS INTEGER</i> <i>N = 4 : S = 0</i> <i>WHILE N <= 8</i> <i>S = S + 15</i> <i>N = N + 1</i> <i>WEND</i> <i>PRINT S</i></pre>	<pre><i>var n, s: integer;</i> <i>begin</i> <i>n := 4; s := 0;</i> <i>while n <= 8 do begin</i> <i>s := s + 15;</i> <i>n := n + 1;</i> <i>end;</i> <i>write(s);</i> <i>end.</i></pre>
Си	Алгоритмический
<pre><i>#include <stdio.h></i> <i>void main(){</i> <i>int n, s;</i> <i>n = 4; s = 0;</i> <i>while (n <= 8){</i> <i>s = s + 15;</i> <i>n = n + 1;</i> <i>}</i> <i>printf(«%d», s);</i> <i>}</i></pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> <i>n, s</i> <i>n := 4; s := 0</i> <u>нц пока</u> <i>n <= 8</i> <i>s := s + 15</i> <i>n := n + 1</i> <u>кц</u> <u>вывод</u> <i>s</i> <u>кон</u></pre>

Ответ: 75

26. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 0 : S = 512 WHILE S >= 0 S = S - 20 N = N + 1 WEND PRINT N </pre>	<pre> var n, s: integer; begin n := 0; s := 512; while s >= 0 do begin s := s - 20; n := n + 1; end; write(n); end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main(){ int n, s; n = 0; s = 512; while (s >= 0){ s = s - 20; n = n + 1; } printf(«%d», n); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> n, s n := 0; s := 512 <u>нц пока</u> s >= 0 s := s - 20 n := n + 1 <u>кц</u> <u>вывод</u> n <u>кон</u> </pre>

Ответ: 26

27. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER </pre>	<pre> var n, s: integer; </pre>

<pre> N = 14 : S = 0 WHILE N <= 18 S = S + 25 N = N + 1 WEND PRINT S </pre>	<pre> begin n := 14; s := 0; while n <= 18 do begin s := s + 25; n := n + 1; end; write(s); end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main(){ int n, s; n = 14; s = 0; while (n <= 18){ s = s + 25; n = n + 1; } printf(«%d», s); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> n, s n := 14; s := 0 <u>нц пока</u> n <= 18 s := s + 25 n := n + 1 <u>кц</u> <u>вывод</u> s <u>кон</u> </pre>

Ответ: 125

28. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 24 : S = 0 WHILE N <= 28 S = S + 20 N = N + 2 </pre>	<pre> var n, s: integer; begin n := 24; s := 0; while n <= 28 do begin s := s + 20; </pre>

<p><i>WEND</i></p> <p><i>PRINT S</i></p>	<p><i>n := n + 2;</i></p> <p><i>end;</i></p> <p><i>write(s);</i></p> <p><i>end.</i></p>
<p>Си</p>	<p>Алгоритмический</p>
<pre><i>#include<stdio.h></i> <i>void main(){</i> <i>int n, s;</i> <i>n = 24; s = 0;</i> <i>while (n <= 28){</i> <i>s = s + 20;</i> <i>n = n + 2;</i> <i>}</i> <i>printf(«%d», s);</i> <i>}</i></pre>	<pre><u><i>алг</i></u> <u><i>нач</i></u> <u><i>цел n, s</i></u> <i>n := 24; s := 0</i> <u><i>нц пока n <= 28</i></u> <i>s := s + 20</i> <i>n := n + 2</i> <u><i>кц</i></u> <u><i>вывод s</i></u> <u><i>кон</i></u></pre>

Ответ: 60

29. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 0 S = 0 WHILE S <= 365 S = S + 37 N = N + 5 WEND PRINT N </pre>	<pre> var n, s: integer; begin n := 0; s := 0; while s <= 365 do begin s := s + 37; n := n + 5 end; write(n) end. </pre>
Сп	Алгоритмический
<pre> #include<stdio.h> void main() { int n, s; n = 0; s = 0; while (s <= 365) { s = s + 37; n = n + 5; } printf("%d", n); } </pre>	<pre> алг нач цел n, s n := 0 s := 0 нц пока s <= 365 s := s + 37 n := n + 5 кц вывод n кон </pre>

Ответ: 50

30. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 1 S = 5 WHILE S <= 365 S = S + 37 N = N * 2 WEND PRINT N </pre>	<pre> var n, s: integer; begin n := 1; s := 5; while s <= 365 do begin s := s + 37; n := n * 2; end; write(n) end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main() { int n, s; n = 1; s = 5; while (s <= 365) { s = s + 37; n = n * 2; } printf("%d", n); } </pre>	<pre> алг нач цел n, s n := 1 s := 5 нц пока s <= 365 s := s + 37 n := n * 2 кц вывод n кон </pre>

Ответ: 1024

31. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования):

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 1 S = 0 WHILE S <= 370 S = S + 37 N = N + 10 WEND PRINT N </pre>	<pre> var n, s: integer; begin n := 1; s := 0; while s <= 370 do begin s := s + 37; n := n + 10; end; write(n) end. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> void main() { int n, s; n = 1; s = 0; while (s <= 370) { s = s + 37; n = n + 10; } printf("%d", n); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> n, s n := 1 s := 0 <u>нц пока</u> s <= 370 s := s + 37 n := n + 10 <u>кц</u> <u>вывод</u> n <u>кон</u> </pre>

Ответ: 111

32. Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующим соотношением:

$$F(n) = F(n-1) + 2 \cdot F(n-2) \text{ при } n > 2$$

$$F(1) = 0$$

$$F(2) = 1$$

Чему равно значение функции $F(6)$? В ответ запишите только натуральное число.

Ответ: 11

33. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
---------------	----------------

<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) F = 9*(x+19)*(x-19)+1; END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 9*(x+19)*(x-19) + 1; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)<R) then begin M:=t; R:=F(t); end; end; write(M); END. </pre>
<p>Си</p>	<p>Алгоритмический</p>
<pre> #include<stdio.h> int F(int x) { return 9*(x+19)*(x-19)+1; } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u>а, b, t, R, M a:= -20; b:= 20 M:= a; R:= F(a) <u>нцдл</u>я t <u>от</u> a <u>до</u> b <u>если</u> F(t)< R <u>то</u> M := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u>M </pre>

<pre> M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>кОН</u> алг цел F(цел x) <u>нач</u> <u>знач</u> := 9*(x+19)*(x-19)+1 <u>кОН</u> </pre>
--	---

Ответ: 0

34. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) F = 19*(x-19)*(x-17)+17; END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 19*(x-19)*(x-17)+17; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)<R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> </pre>	<pre> <u>алг</u> </pre>

<pre> int F(int x) { return 19*(x-19)*(x-17)+17; } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>нач</u> цел a, b, t, R, M a := -20; b := 20 M := a; R := F(a) <u>нц</u> <u>для</u> t <u>от</u> a <u>до</u> b <u>если</u> F(t)< R <u>то</u> M := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u> M <u>кон</u> <u>алг</u> <u>цел</u> F(<u>цел</u> x) <u>нач</u> <u>знач</u> :=19*(x-19)*(x-17)+17 <u>кон</u> </pre>
---	--

Ответ: 18

35. Определите, какое число будет напечатано в результате выполнения следующего алгоритма:

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) F = 19*(x-19)* (x+17)+17; END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 19*(x-19)* (x+17)+17; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)<R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> int F(int x) { return 19*(x-19)* (x+17)+17; } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u>а, b, t, R, M а := -20; b := 20 М := а; R := F(а) <u>нц</u>для t <u>от</u> а <u>до</u> b <u>если</u> F(t) < R <u>то</u> М := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u>М <u>кон</u> <u>алгцел</u> F(<u>цел</u> x) <u>нач</u> <u>знач</u> :=19*(x-19)* (x+17)+17 <u>кон</u> </pre>

Ответ: 1

36. Определите, какое число будет напечатано в результате выполнения следующего алгоритма:

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) r = (x-3)*(x-3) F = r*r; END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin integer r; r:= (x-3)*(x-3); F := r*r; end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)<R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> int F(int x) {int r; r=(x-3)*(x-3); return r*r; } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)<R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>алг</u> <u>нач</u> целa, b, t, R, M a := -20; b := 20 M := a; R := F(a) <u>нц</u> <u>для</u> t <u>от</u> a <u>до</u> b <u>если</u> F(t)< R <u>то</u> M := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u>M <u>кон</u> <u>алг</u> <u>цел</u> F(<u>цел</u> x) <u>нач</u> <u>цел</u> r r:= (x-3)*(x-3) <u>знач</u> :=r*r <u>кон</u> </pre>

Ответ: 3

37. Определите, какое число будет напечатано в результате выполнения следующего алгоритма:

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -10: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) > R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) F = 19*(x-1)*(x-1); END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 19*(x-1)*(x-1); end; BEGIN a := -10; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)>R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> int F(int x) { return 19*(x-1)*(x-1); } void main() { int a, b, t, M, R; a = -10; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)>R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> <u>алг</u> <u>нач</u> целa, b, t, R, M a := -10; b := 20 M := a; R := F(a) <u>нц</u> t <u>от</u> a <u>до</u> b <u>если</u> F(t) > R <u>то</u> M := t; R := F(t) <u>все</u> <u>кц</u> <u>вывод</u> M <u>кон</u> <u>алгцел</u> F(<u>цел</u> x) <u>нач</u> <u>знач</u> := 19*(x-1)*(x-1) <u>кон</u> </pre>

Ответ: 20

38. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -10: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) > R THEN M = T R = F(T) END IF NEXT T PRINT M FUNCTION F(x) F = 19 - 19*(x-1)*(x-1) END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 19 - 19*(x-1)*(x-1); end; BEGIN a := -10; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)>R) then begin M := t; R := F(t); end; end; write(M); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> int F(int x) { return 19 - 19*(x-1)*(x-1); } void main() { int a, b, t, M, R; a = -10; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)>R) { M = t; R = F(t); } } printf("%d", M); } </pre>	<pre> алг нач цел a, b, t, M, R a := -10; b := 20 M := a; R := F(a) нц для t от a до b если F(t) > R то M := t; R := F(t) все кц вывод M кон алг цел F(цел x) нач знач := 19 - 19*(x-1)*(x-1) кон </pre>

Ответ: 1

39. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -10: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) > R THEN M = T R = F(T) END IF NEXT T PRINT R FUNCTION F(x) F = 21 - 19*(x-1)*(x-1); END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x: integer):integer; begin F := 21 - 19*(x-1)*(x-1); end; BEGIN a := -10; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t)>R) then begin M := t; R := F(t); end; end; write(R); END. </pre>
Сп	Алгоритмический
<pre> #include<stdio.h> int F(int x) { return 21 - 19*(x-1)*(x-1); } void main() { int a, b, t, M, R; a = -10; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t)>R) { M = t; R = F(t); } } printf("%d", R); } </pre>	<pre> алг нач цел a, b, t, M, R a := -10; b := 20 M := a; R := F(a) нц для t от a до b если F(t) > R то M := t; R := F(t) все кц вывод R кон алг цел F(цел x) нач знач := 21 - 19*(x-1)*(x-1) кон </pre>

Правильно: **21**

40. Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на четырех языках):

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -20: B = 20 M = A: R = F(A) FOR T = A TO B IF F(T) < R THEN M = T R = F(T) END IF NEXT T PRINT R FUNCTION F(x) F = 19*(x-21)*(x-21); END FUNCTION </pre>	<pre> var a,b,t,M,R :integer; Function F(x:integer):integer; begin F := 19*(x-21)*(x-21); end; BEGIN a := -20; b := 20; M := a; R := F(a); for t := a to b do begin if (F(t) < R) then begin M := t; R := F(t); end; end; write(R); END. </pre>
Си	Алгоритмический
<pre> #include<stdio.h> int F(int x) { return 19*(x-21)*(x-21); } void main() { int a, b, t, M, R; a = -20; b = 20; M = a; R = F(a); for (t=a; t<=b; t++){ if (F(t) < R) { M = t; R = F(t); } } printf("%d", R); } </pre>	<pre> алг нач цел a, b, t, M, R a := -20; b := 20 M := a; R := F(a) нц для t от a до b если F(t) < R то M := t; R := F(t) все кц вывод R кон алг цел F(цел x) нач знач := 19*(x-21)*(x-21) кон </pre>

Ответ: 19

41. **А 12 № 3354.** В программе описан одномерный целочисленный массив *A* с индексами от 0 до 10. Ниже представлен фрагмент этой программы, в котором значения элементов массива сначала задаются, а затем меняются.

$A[0] := 1;$

for $i := 1$ to 10 do

$A[i] := 2 + A[i-1];$

for $i := 0$ to 5 do

$A[10-i] := A[i]-1;$

Чему будут равны элементы этого массива?

- 1) **1 3 5 7 9 10 8 6 4 2 0**
- 2) 1 3 5 7 9 11 13 15 17 19 21
- 3) 1 3 5 7 9 11 9 7 5 3 1
- 4) 1 3 5 7 9 11 8 6 4 3 0

42. В программе описан одномерный целочисленный массив с индексами от 0 до 10 и целочисленные переменные k, i. В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for i := 0 to 10 do A[i] := i;  
for i := 0 to 4 do begin  
k := A[i];  
A[i] := A[10-i];  
A[10-i] := k;  
end;
```

Чему будут равны элементы этого массива?

- 1) **10 9 8 7 6 5 4 3 2 1 0**
- 2) 0 1 2 3 4 5 6 7 8 9 10
- 3) 0 1 2 3 4 5 4 3 2 1 0
- 4) 10 9 8 7 6 5 6 7 8 9 10

43. В программе описан одномерный целочисленный массив с индексами от 0 до 10. В приведенном ниже фрагменте программы массив сначала заполняется, а потом изменяется:

```
for i := 0 to 10 do  
A[i] := i;  
for i := 1 to 11 do  
A[i-1] := A[11-i];
```

Чему будут равны элементы этого массива?

- 1) 10 9 8 7 6 5 4 3 2 1 0
- 2) 11 10 9 8 7 6 5 4 3 2 1
- 3) 10 9 8 7 6 5 6 7 8 9 10**
- 4) 11 10 9 8 7 6 7 8 9 10 11

44. Значения элементов двух массивов А и В размером 1 x 100 задаются с помощью следующего фрагмента программы:

```
for i:=1 to 100 do  
A[i] := 50 - i;  
for i:=1 to 100 do  
B[i] := A[i] + 49;
```

Сколько элементов массива В будут иметь отрицательные значения?

- 1) 1**
- 2) 10
- 3) 50
- 4) 100

45. Значения двух массивов А[1..100] и В[1..100] задаются с помощью следующего фрагмента программы:

```
for n:=1 to 100 do  
A[n] := n - 10;  
for n:=1 to 100 do  
B[n] := A[n]*n;
```

Сколько элементов массива В будут иметь положительные значения?

- 1) 10
- 2) 50
- 3) 90**
- 4) 100

Примерная тематика рефератов, докладов и презентаций:

1. Информационно вычислительная система (ивс).
2. Языки программирования.
3. История развития языка программирования Си.
4. Языки программирования (содержание реферата: машинный код процессора, алгоритм и программа, компиляторы и интерпретаторы, уровни языков программирования, поколения языков программирования).
5. Языки программирования (содержание реферата: языки программирования высокого уровня (Fortran, Cobol, Algol, Pascal, Basic, C, C++, Java).
6. Языки программирования (содержание реферата: языки программирования баз данных, языки программирования для Internet).
7. Алгоритмическое (модульное) программирование (содержание реферата: алгоритм, требования к модулю, переменные, константы, операции, указатели, операторы).
8. Структурное программирование (содержание реферата: подпрограммы, программирование “сверху вниз”, процедуры и функции, событийно-ориентированное программирование).
9. Проектирование программ (содержание реферата: программирование как вид деятельности, экономические аспекты программирования, этапы разработки программ, стандарты качества программного).
10. Коллективная разработка программных продуктов.

2.2. Оценочные средства для промежуточной аттестации

Перечень вопросов для подготовки к зачету:

Тема: Понятие алгоритма и его свойства.

1. Понятие алгоритма.

2. Понятие блок-схемы.
3. Основные элементы блок-схем.
4. Правила построения блок-схем.

Тема: Основы программирования на языке Си.

1. Начальные сведения о языке программирования Си.
2. Алфавит языка Си.
3. Типы данных.
4. Переменные и константы.
5. Преобразование типов.
6. Ввод и вывод данных.

Тема: Линейный вычислительный процесс.

1. Понятие оператора.
2. Классификация операторов в языке программирования Си.
3. Простой оператор в языке Си.
4. Составной оператор в языке Си.
5. Метка оператора в языке Си.
6. Точка с запятой пустого оператора в языке Си.
7. Унарные операторы в языке Си.
8. Бинарные операторы в языке Си.
9. Комментарии в языке Си.
10. Основные математические операторы.

Тема: Операторы управления. Алгоритмы ветвления.

1. Выражение и операторы.
2. Построение условий.
3. Алгоритмы разветвления.
4. Оператор ветвления if.
5. Оператор ветвления switch.

Тема: Операторы управления. Циклические алгоритмы.

1. Оператор цикла for.
2. Оператор цикла while.
3. Оператор цикла do.
4. Оператор break.
5. Оператор continue.

Тема: Производные типы. Одномерные массивы.

1. Понятие массива.
2. Определение одномерного массива.
3. Обращение к элементам одномерного массива.
4. Типовые задачи по обработке одномерных массивов.

Тема: Производные типы. Двумерные массивы.

1. Понятие двумерного массива.
2. Определение двумерного массива.
3. Обращение к элементам двумерного массива.
4. Типовые задачи по обработке двумерных массивов.

Тема: Указатели.

1. Понятие указателя.
2. Адресные операции.
3. Адресная арифметика.
4. Динамическое распределение памяти.
5. Связь между указателями и массивами.

Тема: Функции.

1. Определение функции.
2. Описание функции.

3. Управление видимости функций.
4. Вызов функций.
5. Передача параметров.
6. Передача массивов в качестве параметров.

3. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОГРАММЫ ДИСЦИПЛИНЫ.

3.1. Текущий контроль успеваемости студентов

Текущий контроль успеваемости – это установление уровня знаний, умений, владений студентов по отношению к объему и содержанию разделов (модулей, частей) учебных дисциплин, представленных и утвержденных в учебных планах и учебных программах.

Текущий контроль успеваемости осуществляется через комплекс испытаний студентов в виде устных и письменных опросов, коллоквиумов, контрольных работ, проверки домашних заданий, защиты отчетов, компьютерного и бланочного тестирования. Возможны и другие виды контроля по усмотрению кафедры, обеспечивающей учебный процесс по данной дисциплине, в том числе, контроль посещаемости занятий.

В систему текущего контроля рекомендуется вводить необязательные мероприятия, позволяющие повысить семестровый рейтинг, например, участие в олимпиадах, научное исследование, участие в научных конференциях с докладом по теме изучаемого предмета и т.д. с назначением определенных баллов, прибавляемых к семестровому рейтингу по дисциплине. При этом рейтинг не должен превышать 100 баллов.

Для текущего контроля успеваемости на кафедрах, осуществляющих учебный процесс, создаются и периодически актуализируются банки тестов, заданий, программы компьютерных проверок и т.п. материалы.

Виды и сроки проведения мероприятий текущего контроля устанавливаются рабочей программой учебной дисциплины.

3.2. Промежуточная аттестация

Промежуточная аттестация студентов – это установление уровня знаний, умений, владений обучаемых, как показателя уровня освоения требуемых компетенций, по отношению к объему и содержанию семестровых частей учебных дисциплин или дисциплин в целом.

Оценка промежуточной аттестации студента по дисциплине формируется на основании семестрового рейтинга текущего контроля и рейтинга зачетного и/или экзаменационного испытания.

Зачетное/экзаменационное испытание проводится в сроки, устанавливаемые в соответствии с утвержденными учебными планами, календарными учебными графиками, приказами.

Преподаватель имеет право принять у студента зачет и/или экзамен только при наличии первичных документов по учету результатов промежуточной аттестации. Первичными документами являются экзаменационные и зачетные ведомости, индивидуальные разрешения на сдачу зачетов, экзаменов, курсовых проектов (работ). Все первичные документы должны передаваться в деканат преподавателем лично не позднее следующего дня после проведения испытания промежуточной аттестации.

По результатам промежуточной аттестации студенту, кроме итогового рейтинга по 100-балльной шкале, выставляется итоговая отметка, которая может быть дифференцированной («отлично», «хорошо», «удовлетворительно», «неудовлетворительно»), либо недифференцированной («зачтено», «не зачтено»).

При аттестации на «отлично», «хорошо», «удовлетворительно» и «зачтено» студент считается получившим положительную оценку и прошедшим промежуточную аттестацию. Положительные оценки и соответствующие рейтинги заносятся в первичные документы и зачетные книжки студентов. Записи в зачетных книжках студентов должны осуществляться только после оформления первичных документов.

Оценки «неудовлетворительно» и «не зачтено» проставляются только в первичные документы.

Неудовлетворительные результаты промежуточной аттестации по одному или нескольким учебным курсам, дисциплинам (модулям) образовательной программы или непрохождение промежуточной аттестации в установленные сроки признаются академической задолженностью. Студенты обязаны ликвидировать академическую задолженность.

Виды и сроки проведения мероприятий промежуточной аттестации устанавливаются рабочей программой учебной дисциплины.