Перечень теоретических и практических заданий к экзамену
по ОП.08 Теория алгоритмов
(2 курс, 3 семестр 2022-2023 уч. г.)

Форма контроля: Контрольная работа (Опрос)
Описательная часть: экзаменационный билет содержит одно теоретическое и одно практическое задание

Перечень теоретических заданий:

Задание №1

Дать определение "**Дискретность**".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №2

Дать определение "**Формальность**".

Оценка	Показатели оценки

5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №3

Дать определение "**Результативность**".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.

4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №4

Дать определение "**Конечность**".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №5

Дать определение "**Массовость**".

Оценка	Показатели оценки

5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №6

Дать определение "**Язык программирования**".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.

4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №7

Дать определение "**Алгоритм Евклида**".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №8

Дать определение "***Алгоритм***".

Оценка	Показатели оценки

5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №9

Дать определение "**Однозначность** ".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.

4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №10

Дать определение "**Алфавит языка** ".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №11

Дать определение "**Синтаксис** ".

Оценка	Показатели оценки

5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №12

Дать определение "**Семантика** ".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.

4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №13

Дать определение "**Тип данных** ".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №14

Дать определение "**Описание данных**".

Оценка	Показатели оценки

5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №15

Дать определение "**Переменная** ".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.

4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №16

Дать определение "**Целый тип** ".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №17

Дать определение "**Вещественный тип** ".

Оценка	Показатели оценки

5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №18

Дать определение "**Арифметические выражения** ".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.

4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №19

Дать определение "**Оператор присваивания** ".

Оценка	Показатели оценки
5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №20

Дать определение "**Логический тип** ".

Оценка	Показатели оценки

5	Дано полное формальнологическое определение термину. Пояснено и приведены примеры применения термина. Образец ответа: Детерминированность (от лат. determinans — определяющий) — определяемость. Детерминированность может подразумевать определяемость на общегносеологическом уровне или для конкретного алгоритма. Под жесткой детерминированностью процессов в мире понимается однозначная предопределенность, т. е. у каждого следствия есть строго определенная причина. В таком смысле является антонимом стохастичности. Но детерминированность не всегда тождественна предопределенности. Например, может быть детерминированность будущим (целевая детерминация), когда предполагаемые субъектом цели в его возможном будущем определяют его поведение в настоящем.
4	Дано полное формальнологическое определение термину. Приведены примеры.
3	Дано определение термину.

Задание №21

Описать метод вычисления сложности работы алгоритма:

```
var n: integer; a,b,c: byte;begin randomize; n := random(900) + 100; w
riteln(n); a := n div 100; b := n div 10 mod 10; c := n mod 10; writel
n('?????: ',a+b+c); writeln('???????????: ',a*b*c);end.
```

Оценка	Показатели оценки

5	Обучающийся должен: I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать каждый метод: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик) 2) Сложность алгоритма по памяти: • память состоит из ячеек, каждая из которых имеет адрес и может хранить один элемент данных; • каждое обращение к памяти занимает одну единицу времени, независимо от номера адресуемой ячейки; • количество памяти достаточно для выполнения любого алгоритма; • процессор выполняет любую элементарную операцию за один временной шаг; • циклы и функции не считаются элементарными операциями.
4	Обучающийся должен I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать методом: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик)
3	Обучающийся должен указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов.

Задание №22

Описать метод вычисления сложности работы алгоритма:

```
var  x1,y1,x2,y2: real; k, b: real; begin write('A(x1;y1): '); readln(
x1, y1); write('B(x2;y2): '); readln(x2, y2);  k := (y1 - y2) / (x1 -
x2); b := y2 - k * x2;  writeln('y = ',k:0:2,'x + ',b:0:2);end.
```

Оценка	Показатели оценки
5	Обучающийся должен: I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать каждый метод: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик) 2) Сложность алгоритма по памяти: • память состоит из ячеек, каждая из которых имеет адрес и может хранить один элемент данных; • каждое обращение к памяти занимает одну единицу времени, независимо от номера адресуемой ячейки; • количество памяти достаточно для выполнения любого алгоритма; • процессор выполняет любую элементарную операцию за один временной шаг; • циклы и функции не считаются элементарными операциями.

4	Обучающийся должен I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать методом: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик)
3	Обучающийся должен указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов.

Задание №23

Описать метод вычисления сложности работы алгоритма:

```
var n: integer; begin    randomize;     n := random(900)+100;     if n
 mod 10 <> 0 then  n := n - n mod 10;  writeln(n); end.
```

Оценка	Показатели оценки

5	Обучающийся должен: I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать каждый метод: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик) 2) Сложность алгоритма по памяти: • память состоит из ячеек, каждая из которых имеет адрес и может хранить один элемент данных; • каждое обращение к памяти занимает одну единицу времени, независимо от номера адресуемой ячейки; • количество памяти достаточно для выполнения любого алгоритма; • процессор выполняет любую элементарную операцию за один временной шаг; • циклы и функции не считаются элементарными операциями.
4	Обучающийся должен I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать методом: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик)
3	Обучающийся должен указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов.

Задание №24

Описать метод вычисления сложности работы алгоритма:

```
var a, b: integer;begin writeln('??????? ???? ?????? ? ???? ???????? ? ????'); readln(a, b); if a mod 2 = 0 then  writeln(b,' - ???????? ???? ?') else  writeln(a,' - ???????? ?????');end.
```

Оценка	Показатели оценки
5	Обучающийся должен: I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать каждый метод: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик) 2) Сложность алгоритма по памяти: • память состоит из ячеек, каждая из которых имеет адрес и может хранить один элемент данных; • каждое обращение к памяти занимает одну единицу времени, независимо от номера адресуемой ячейки; • количество памяти достаточно для выполнения любого алгоритма; • процессор выполняет любую элементарную операцию за один временной шаг; • циклы и функции не считаются элементарными операциями.

4	Обучающийся должен I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать методом: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик)
3	Обучающийся должен указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов.

Задание №25

Описать метод вычисления сложности работы алгоритма:

```
var  n: char;begin write('Think of a number of from 1 to 5 and press E
nter'); readln;  write('The number more then 3? (y/n) '); readln(n); i
f n = 'y' then begin  write('The number is 4? (y/n) '); readln(n);  if
 n = 'n' then   writeln('The number is 5.'); end else begin  write('Th
e number is 1? (y/n) '); readln(n);  if n = 'n' then begin   write('Th
e number is 2? (y/n) '); readln(n);   if n = 'n' then    writeln('The
number is 3.');  end; end; writeln('Guessed!');end.
```

Оценка	Показатели оценки

5	Обучающийся должен: I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать каждый метод: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик) 2) Сложность алгоритма по памяти: • память состоит из ячеек, каждая из которых имеет адрес и может хранить один элемент данных; • каждое обращение к памяти занимает одну единицу времени, независимо от номера адресуемой ячейки; • количество памяти достаточно для выполнения любого алгоритма; • процессор выполняет любую элементарную операцию за один временной шаг; • циклы и функции не считаются элементарными операциями.
4	Обучающийся должен I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать методом: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик)
3	Обучающийся должен указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов.

Задание №26

Описать метод вычисления сложности работы алгоритма:

```
var    n: integer; begin    write('n = '); readln(n);     if n = 0 the
n  write('Zero') else begin  if n > 0 then   write('Positive ')  else
   write('Negative ');  if abs(n) < 10 then   write('single')  else
 if (abs(n) >= 10) and (abs(n) < 100) then    write('two-
digit')   else    write('three-digit or more'); end; writeln;end.
```

Оценка	Показатели оценки
5	Обучающийся должен: I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать каждый метод: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик) 2) Сложность алгоритма по памяти: • память состоит из ячеек, каждая из которых имеет адрес и может хранить один элемент данных; • каждое обращение к памяти занимает одну единицу времени, независимо от номера адресуемой ячейки; • количество памяти достаточно для выполнения любого алгоритма; • процессор выполняет любую элементарную операцию за один временной шаг; • циклы и функции не считаются элементарными операциями.

4	Обучающийся должен I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать методом: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик)
3	Обучающийся должен указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов.

Задание №27

Описать метод вычисления сложности работы алгоритма:

```
var    year: word; begin    write('Input year: '); readln(year);     i
f year mod 4 = 0 then  if (year mod 100 = 0) and (year mod 400 <> 0) t
hen   year := 365  else   year := 366 else  year := 365;  writeln('The
 number of days in the year: ', year);end.
```

Оценка	Показатели оценки

5	Обучающийся должен: I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать каждый метод: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик) 2) Сложность алгоритма по памяти: • память состоит из ячеек, каждая из которых имеет адрес и может хранить один элемент данных; • каждое обращение к памяти занимает одну единицу времени, независимо от номера адресуемой ячейки; • количество памяти достаточно для выполнения любого алгоритма; • процессор выполняет любую элементарную операцию за один временной шаг; • циклы и функции не считаются элементарными операциями.
4	Обучающийся должен I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать методом: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик)
3	Обучающийся должен указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов.

Задание №28

Описать метод вычисления сложности работы алгоритма:

```
var    x, y, r: real;    r_xy: real; begin    write('x = '); readln(x)
;    write('y = '); readln(y);    write('R = '); readln(r);     r_xy :
= sqrt(x*x + y*y);     if r_xy <= r then  writeln('The point belongs t
o the circle.') else  writeln('The point does not belong to the circle
.');end.
```

Оценка	Показатели оценки
5	Обучающийся должен: I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать каждый метод: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик) 2) Сложность алгоритма по памяти: • память состоит из ячеек, каждая из которых имеет адрес и может хранить один элемент данных; • каждое обращение к памяти занимает одну единицу времени, независимо от номера адресуемой ячейки; • количество памяти достаточно для выполнения любого алгоритма; • процессор выполняет любую элементарную операцию за один временной шаг; • циклы и функции не считаются элементарными операциями.

4	Обучающийся должен I. Указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов. II. Описать методом: 1) Вычислительная сложность - *подсчет количества выполняемых операций*. Точное количество операций будет зависеть от обрабатываемых данных, поэтому имеет смысл говорить о наилучшем, наихудшем и среднем случаях. (привести примеры расчета, достаточно - счетчик)
3	Обучающийся должен указать в виде оценивающего ресурса процессорное время (*вычислительная сложность*) и память (*сложность алгоритма по памяти*). Оценка позволяет предсказать время выполнения и сравнивать эффективность алгоритмов.

Перечень практических заданий:
Задание №1

Разработать программный код для следующей задачи:

В зрительном зале 25 рядов. В каждом ряду 36 мест (кресел). Информация о проданных билетах хранится в двумерном массиве (номер строки – номер ряда, номер столбца – номер места). Если билет продан значение массива = 1, если нет, то значение= 0. Места распределите случайным образом и составьте программу, определяющую число проданных мест в 12-м ряду

После вывода ответа на вторую задачу поменяйте местами второй и предпоследний столбец местами.

Оценка	Показатели оценки
3	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными.

4	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными. Алгоритм решения задачи соответствует понятиям работы с двумерными массивами.
5	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными. Алгоритм решения задачи соответствует понятиям работы с двумерными массивами. Алгоритм решения задачи соответствует понятиям работы с памятью.

Задание №2

Разработать программный код для следующей задачи:

1. Создать одномерный массив из 20 чисел со значениями в диапазоне [-7;23].

В созданном массиве найти максимальное значение и вывести на экран.

Оценка	Показатели оценки
3	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными.
4	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными. Алгоритм решения задачи соответствует понятиям работы с двумерными массивами.
5	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными. Алгоритм решения задачи соответствует понятиям работы с двумерными массивами. Алгоритм решения задачи соответствует понятиям работы с памятью.

Задание №3

Разработать программный код для следующей задачи:

1. Дан двумерный массив размером 5х6, значения вводятся с клавиатуры.

В первом его столбце найти:

а) сумму нечетных элементов;

б) количество положительных элементов

Оценка	Показатели оценки
3	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными.
4	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Блок-схема соответствует выбранному алгоритму решения задачи.
5	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Алгоритм решения задачи соответствует математическому понятию четных и нечетных чисел. Блок-схема построена в соответствии с ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем»; Блок-схема соответствует выбранному алгоритму решения задачи.

Задание №4

Разработать программный код для следующей задачи:

1. Создать одномерный массив из 30 чисел со значениями в диапазоне [-20;40).

В созданном массиве найти минимальное значение и вывести на экран.

Оценка	Показатели оценки
3	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными.

4	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Блок-схема соответствует выбранному алгоритму решения задачи.
5	Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Алгоритм решения задачи соответствует математическому понятию четных и нечетных чисел. Блок-схема построена в соответствии с ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем»; Блок-схема соответствует выбранному алгоритму решения задачи.

Задание №5

Разработать программный код для следующей задачи:

1. Создать одномерный массив из 15 чисел со значениями в диапазоне

[-1;20).

В созданном массиве найти индекс числа с минимальным значение и вывести на экран.

Оценка	Показатели оценки
3	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными.
4	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Блок-схема соответствует выбранному алгоритму решения задачи.

5	Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Алгоритм решения задачи соответствует математическому понятию четных и нечетных чисел. Блок-схема построена в соответствии с ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем»; Блок-схема соответствует выбранному алгоритму решения задачи.

Задание №6

Разработать программный код для следующей задачи:

1. В поезде 18 вагонов, в каждом из которых 36 мест.

Информация о проданных на поезд билетах хранится в двумерном массиве,

номера строк которых соответствуют номерам вагонов, а номера столбцов — номерам мест.

Если билет на то или иное место продан, то соответствующий элемент массива имеет значение 1,

в противном случае — 0. Составить программу, определяющую число свободных мест в любом из вагонов поезда.

Оценка	Показатели оценки
3	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными.
4	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Блок-схема соответствует выбранному алгоритму решения задачи.

5	Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Алгоритм решения задачи соответствует математическому понятию четных и нечетных чисел. Блок-схема построена в соответствии с ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем»; Блок-схема соответствует выбранному алгоритму решения задачи.

Задание №7

Разработать программный код для следующей задачи:

1. В двумерном массиве хранится информация о зарплате 20 человек за каждый месяц года

(первого человека — в первой строке, второго — во второй и т. д.). Минимальная заработная плата 12 тыс. рублей., максимум 80тыс.

Составить программу для расчета общей зарплаты, полученной за год любым человеком, информация о зарплате которого представлена в массиве.

Оценка	Показатели оценки
3	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными.
4	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Блок-схема соответствует выбранному алгоритму решения задачи.
5	Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Алгоритм решения задачи соответствует математическому понятию четных и нечетных чисел. Блок-схема построена в соответствии с ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем»; Блок-схема соответствует выбранному алгоритму решения задачи.

Задание №8

Разработать программный код для следующей задачи:

1. Дан двумерный массив размером 10х15. Значения вводятся с клавиатуры.

Определить:

а) сумму всех элементов массива;

б) сумму квадратов всех элементов массива;

в) среднее арифметическое всех элементов массива.

Оценка	Показатели оценки
3	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными.
4	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Блок-схема соответствует выбранному алгоритму решения задачи.
5	Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Алгоритм решения задачи соответствует математическому понятию четных и нечетных чисел. Блок-схема построена в соответствии с ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем»; Блок-схема соответствует выбранному алгоритму решения задачи.

Задание №9

Разработать программный код для следующей задачи:

1. Дан квадратный двумерный массив, размер которого определяется пользователем, а значения случайны в диапазоне [-10;10) .

Определить:

а) сумму квадратов элементов четвертого столбца массива;

б) сумму квадратов элементов *k*-й строки массива.

Оценка	Показатели оценки
3	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными.
4	Необходимо выполнить следующие операции и приемы: Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Блок-схема соответствует выбранному алгоритму решения задачи.
5	Описание входных данных (их типов, диапазонов); Описание выходных данных; Запись математического соотношения, связывающего результат с исходными данными; Алгоритм решения задачи соответствует математическому понятию четных и нечетных чисел. Блок-схема построена в соответствии с ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем»; Блок-схема соответствует выбранному алгоритму решения задачи.

Задание №10

Дан программный код, определить сложность работы алгоритма:

```
min := array[1]
for i:= 2 to N do
  if array[i] < min then min := array[i];
write(min);
```

Оценка	Показатели оценки
3	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* с использованием таблицы определения сложности алгоритма.
4	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.
5	Рассчитана сложность работы алгоритма двумя методами (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.

Задание №11

Дан программный код, определить сложность работы алгоритма:

```
var     n: longint;    sum: integer;begin    readln(n);       sum := 0
;    while n > 0 do begin        if n mod 10 mod 2 = 0 then
 sum := sum + n mod 10;        n := n div 10;    end;        writeln(s
um);end.
```

Оценка	Показатели оценки
3	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* с использованием таблицы определения сложности алгоритма.
4	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.
5	Рассчитана сложность работы алгоритма двумя методами (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.

Задание №12

Дан программный код, определить сложность работы алгоритма:

```
const     N = 20;var     a: array[1..N] of integer;    i: byte;begin
  randomize;    for i:=1 to N do begin        a[i] := random(100);
    write(a[i]:4);    end;    writeln;    writeln('???????? ?????? ???
????????: ');    for i:=1 to N-1 do        if a[i] < a[i+1] then
       write(a[i+1]:4);    writeln;end.
```

Оценка	Показатели оценки
3	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* с использованием таблицы определения сложности алгоритма.
4	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.
5	Рассчитана сложность работы алгоритма двумя методами (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.

Задание №13

Дан программный код, определить сложность работы алгоритма:

```
const    M = 10;var    C: array[1..M] of word;    maxC: word;    i: by
```

```
te; begin    randomize;    maxC := 0;    write('???????? ??????: ');
  for i := 1 to M do begin        C[i] := random(1000);        write (
C[i]:4);        if maxC < C[i] then            maxC := C[i];    end;
   writeln;    writeln('????????: ', maxC);    write('???????: ');
for i := 1 to M do begin        write (C[i]/maxC:6:2);    end;    writ
eln;end.
```

Оценка	Показатели оценки
3	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* с использованием таблицы определения сложности алгоритма.
4	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.
5	Рассчитана сложность работы алгоритма двумя методами (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.

Задание №14

Дан программный код, определить сложность работы алгоритма:

```
const N = 20;var     arr: array[1..N] of integer;    i: byte;begin
randomize;    for i:=1 to N do begin        arr[i] := random(100) - 75
;        write(arr[i]:4);    end;    writeln;    for i:=1 to N do
   if arr[i] > 0 then begin            writeln(i, ' ', arr[i]);
     break;        end;end.
```

Оценка	Показатели оценки
3	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* с использованием таблицы определения сложности алгоритма.
4	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.
5	Рассчитана сложность работы алгоритма двумя методами (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.

Задание №15

Дан программный код, определить сложность работы алгоритма:

```
const N = 10;var    arr: array[1..N,1..N] of integer;    max: integer;
```

```
    i,j: byte;    begin    randomize;    for i:=1 to N do begin
 for j:=1 to N do begin            arr[i,j] := random(1000);
  write(' |',arr[i,j]:3,'| ');        end;        writeln;    end;
for i:=1 to N do         write(' ----- ');    writeln;    for j:=1 to
N do begin        max := arr[1,j];        for i:=2 to N do
if arr[i,j] > max then               max := arr[i,j];        write('
 ',max:3,'  ');    end;    writeln;end.
```

Оценка	Показатели оценки
3	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* с использованием таблицы определения сложности алгоритма.
4	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.
5	Рассчитана сложность работы алгоритма двумя методами (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.

Задание №16

Дан программный код, определить сложность работы алгоритма:

```
var    str1,str2,str3: string;    l,i: byte;    begin    readln(str1);
    str2 := '';    str3 := '';    l := length(str1);    for i:=1 to l
do        if odd(i) then            str3 := str3 + str1[i]        else
            str2 := str2 + str1[i];    writeln(str2);    writeln(str3)
;end.
```

Оценка	Показатели оценки
3	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* с использованием таблицы определения сложности алгоритма.
4	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.
5	Рассчитана сложность работы алгоритма двумя методами (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.

Задание №17

Дан программный код, определить сложность работы алгоритма:

```
const N = 10;var    arr: array[1..N] of integer;    i: byte;begin    r
andomize;    for i:=1 to N do begin        arr[i] := random(100) - 50;
        write(arr[i]:4);    end;    writeln;    for i:=1 to N do
   arr[i] := -1 * arr[i];    for i:=1 to N do        write(arr[i]:4);
    writeln;end.
```

Оценка	Показатели оценки
3	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* с использованием таблицы определения сложности алгоритма.
4	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.
5	Рассчитана сложность работы алгоритма двумя методами (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.

Задание №18

Дан программный код, определить сложность работы алгоритма:

```
const     N = 7; M = 5;var    arr: array[1..N,1..M] of byte;    i,j,a,
b,buff: byte;begin    randomize;    for i:=1 to N do begin        for
j:=1 to M do begin            arr[i,j] := random(20);            write
(arr[i,j]:3);        end;        writeln;    end;    write('????? ????
?? ????????: ');    readln(a,b);    for j:=1 to M do begin        buff
 := arr[a,j];        arr[a,j] := arr[b,j];        arr[b,j] := buff;
 end;    for i:=1 to N do begin        for j:=1 to M do            wri
te(arr[i,j]:3);        writeln;    end;end.
```

Оценка	Показатели оценки
3	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* с использованием таблицы определения сложности алгоритма.
4	Рассчитана сложность работы алгоритма одним из методов (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.
5	Рассчитана сложность работы алгоритма двумя методами (*вычислительная сложность* или *сложность алгоритма по памяти)* без использования таблицы определения сложности алгоритма.

Задание №19

Привести примеры алгоритмов Евклида.

Оценка	Показатели оценки
5	Привести три примера алгоритма Евклида
4	Привести два примера алгоритма Евклида
3	Привести один пример алгоритма Евклида

Задание №20

Почему необходимо формальное определение алгоритма.

Оценка	Показатели оценки
5	Привести три примера.
4	Привести два примера.
3	Привести один пример.