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 два практических задания

Перечень теоретических заданий:
Задание №1

Решите тест из 20 вопросов.

Каждый правильный ответ на вопрос оценивается в один балл. Возможен только один правильный ответ в каждом вопросе. Максимальное количество баллов - 20.

1. Что такое алгоритм?

   набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное число действий.

   инсталляция порядка действий для ЭВМ, который составлен для определенного языка программирования и удовлетворяет определенным правилам.

   инсталляция порядка действий для ЭВМ, который составлен и удовлетворяет определенным правилам

   искусство счета с помощью цифр, но поначалу слово «цифра» относилось только к нулю

2. Какое из перечисленных свойств не относится к свойствам алгоритмов?

   Дискретность Определенность Результативность Полнота Массовость Правильность

3. Какое свойство алгоритмов характеризует данное определение: «Алгоритм должен приводить к решению за конечное число шагов»

   Дискретность Определенность Результативность Полнота Массовость Правильность

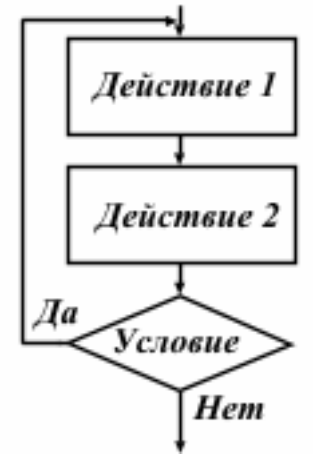
4. Какое свойство алгоритмов характеризует данное определение: «Каждое правило алгоритма должно быть четким, однозначным»

   Дискретность Определенность Результативность Полнота Массовость Правильность

5. Какое свойство алгоритмов характеризует данное определение: «Алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными»

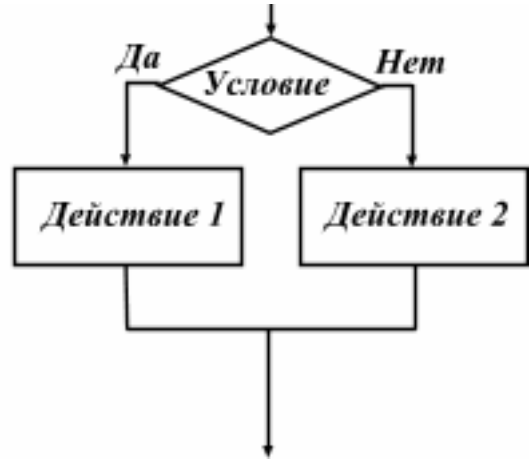
   Дискретность Определенность Результативность Полнота Массовость Правильность

6. Какой вид алгоритма представлен на рисунке?



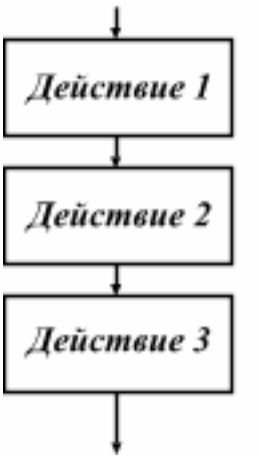
   Линейный Разветвляющий Циклический

7. Какой вид алгоритма представлен на рисунке?



   Линейный Разветвляющий Циклический

8. Какой вид алгоритма представлен на рисунке?



Линейный Разветвляющий Циклический

9. Выберите вариант, где представлены все способы описания алгоритма (но нет лишних).

   Словесный, Формульно-словесный, Графический, Программный

   Словесный, Формульный, Формульно-словесный, Графический, Программный

   Словесный, Формульный, Графический, Программный

   Формульный, Графический, Программный

10. Что такое блок схема?

    распространенный тип схем, описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности.

    схематичное изображение элементов алгоритма, которые собранны воедино с помощью линий в определенном порядке, демонстрируя алгоритм.

    строгое изображение элементов алгоритма, которые собранны воедино по правилам с помощью линий в определенном порядке, демонстрируя алгоритм.

    схема программы на языке программирования, представленная в виде блоков, соединенных линиями

11. Блок схема. На рисункепредставлено обозначение элемента алгоритма. Выберите его наименование



Блок начало-конец

Блок действия

Логический блок (блок условия)

Данные

12. Блок схема. На рисунке представлено обозначение элемента алгоритма. Выберите его наименование



Блок начало-конец

Блок действия

Логический блок (блок условия)

Данные

13. Блок схема. На рисунке представлено обозначение элемента алгоритма. Выберите его наименование



Блок начало-конец

Блок действия

Логический блок (блок условия)

Данные

14. Блок схема. На рисунке представлено обозначение элемента алгоритма. Выберите его наименование



Блок начало-конец

Блок действия

Логический блок (блок условия)

Данные

15. В чем заключается Разработка алгоритма решения задачи

    в установлении необходимой последовательности арифметических и логических действий, строгое выполнение которых приводит к решению задачи.

    строгое выполнение арифметических и логических задач в необходимой последовательности в программном коде

    выполнение арифметических и логических задач в необходимой последовательности в программном коде несет рекомендательный характер

16. Вставьте пропущенное слово: « Любой алгоритм применяется к … и выдает результат»

    исходным данным

    внешним данным

    используемым средствам

    внутренним средствам

17. Вставьте пропущенное слово: «Данные для своего размещения требуют …. … обычно считается дискретной. Единицы измерения … и данных должны быть согласованы между собой»

    память

    элементарный шаг

    участок

    промежуток

18. Вставьте пропущенное слово: «Алгоритм состоит из отдельных …»

    Участков памяти

    элементарный шагов

    участков

    промежутков

19. Вставьте пропущенное слово: «Последовательность шагов алгоритма …, т.е. после каждого шага указывается следующий шаг, либо алгоритм останавливается»

    детерминирована

    результативна

    объективна

    полна

20. Вставьте пропущенное слово: «Каждый алгоритм должен быть … , т.е. после конечного числа шагов выдавать результат»

    детерминировным

    результативным

    объективным

    полным

Оценка	Показатели оценки
3	Обучающийся ответил на 6-13 вопросов
4	Обучающийся ответил на 14-17 вопросов
5	Обучающийся ответил на 18-20 вопросов

Задание №2

Выполните 17 теоретических заданий (в трех первых заданиях ответ - действительные числа, в остальных 14 - выбор ответа).

Каждое правильно выполненное задание оценивается от 1 до 3 баллов (количество баллов указано в формулировке каждого задания). Возможен только один правильный ответ в каждом задании. Максимальное количество баллов - 22.

1. Определите, что будет напечатано в результате выполнения программы (2 балла)

```
var n, s: integer;
begin
  n := 0;
  s := 512;
  while s >= 0 do
  begin
    s := s - 20;
    n := n + 1
  end;
  write(n)
end.
```

2. Получив на вход число x, этот алгоритм печатает два числа *a* и *b*. Укажите НАИБОЛЬШЕЕ из таких чисел x, при вводе которых алгоритм печатает сначала 2, а потом 8 (3 балла)

```
var x, a, b: integer;
begin
  readln(x);
  a:=0; b:=0;
  while x>0 do
  begin
    a:=a+1;
    b:=b+(x mod 10);
    x:=x div 10;
  end;
  writeln(a); write(b);
end.
```

3. Определите, что будет напечатано в результате выполнения программы (2 балла)

```
var n, s: integer;
begin
  n := 24;
  s := 0;
  while n <= 28 do
  begin
    s := s + 20;
    n := n + 2
  end;
  write(s)
end.
```

4. Задача: даны две цифры, необходимо узнать какая из них больше и вывести ответ на экран. Какого вида алгоритма будет достаточно для решения данной задачи? (1 балл)

   Линейного

   Разветвляющего

   Циклического

5. Что такое Цикл (1 балл)

   - это последовательность операторов, которая может выполняться один или более раз

   - это последовательность переменных, которые выполняются один или более раз

   - это последовательность переменных, которые выполняются хаотичное число раз

   - это последовательность операторов, которая может выполняться хаотичное раз

6. К какому виду цикла относится данное определение: «используются тогда, когда не известно, сколько раз нужно выполнить тело цикла (while)» (1 балл)

   Цикл с предусловием

   Цикл с постусловием

   Цикл с параметром (счетчик)

Вложенные циклы

7. К какому виду цикла относится данное определение: «используются тогда, когда не известно, сколько раз нужно выполнить тело цикла (repeat)» (1 балл)

   Цикл с предусловием

   Цикл с постусловием

   Цикл с параметром (счетчик)

   Вложенные циклы

8. К какому виду цикла относится данное определение: «используется тогда, когда заранее определенное число повторений» (1 балл)

   Цикл с предусловием

   Цикл с постусловием

   Цикл с параметром (счетчик)

   Вложенные циклы

9. К какому виду цикла относится данное определение: «находится в теле другого цикла» (1 балл)

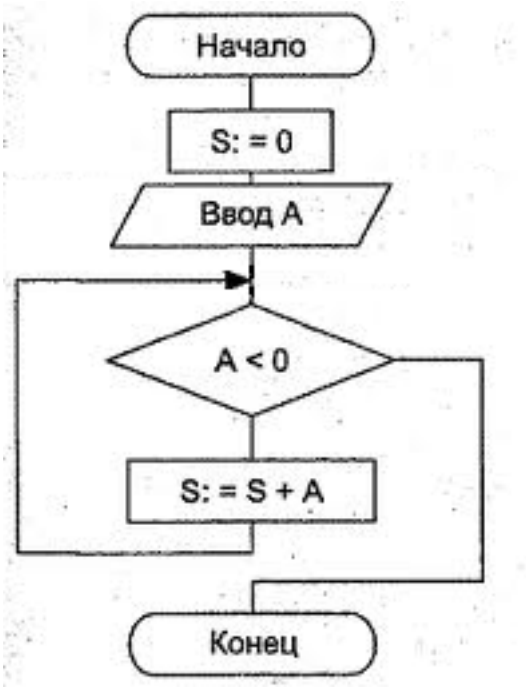
   Цикл с предусловием

   Цикл с постусловием

   Цикл с параметром (счетчик)

   Вложенные циклы

10. На Картинке представлена блок-схема. Выберите код данной блок-схемы. (1 балл)




```
Program;
var S;A:=integer;
begin
S:=0;
readln(A);
while A<0 do
begin
S:=S+A;
end;
end.
```

```
Program;
var S;A:=integer;
begin
readln(A);
S:=0;
while A<0 do
begin
S:=S+A;
end;
end.
```

```
Program;
var S;A:=integer;
begin
S:=0;
```

```
readln(A);
repeat
S:=S+A; until A<0
end;
end.
```

```
Program;
var S;A:=integer;
begin
readln(A);
S:=0;
repeat
S:=S+A; until A<0
end;
end.
```

11. На картинке представлены блок-схемы, выберете нужную по данной программе: (2 балла)

```
program ;

Var

  n,k,p:longint;

Begin

Write('Введите число N > 0, N = '); readln(n);

k:=0;

while p < n Do

Begin

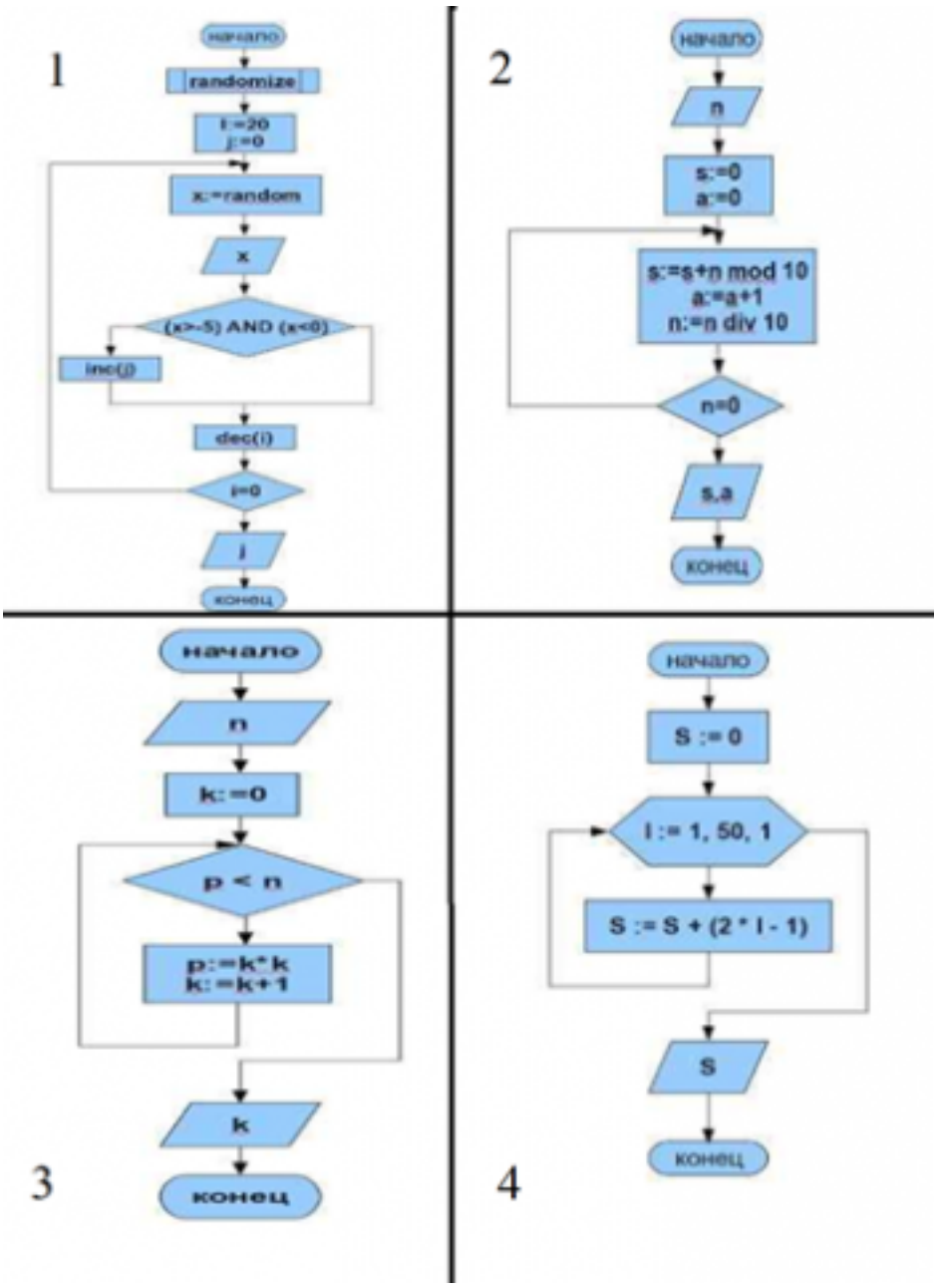
  p := k * k;

  k := k + 1;

end;

writeln(', n, ' равен K = ',k);

End.
```



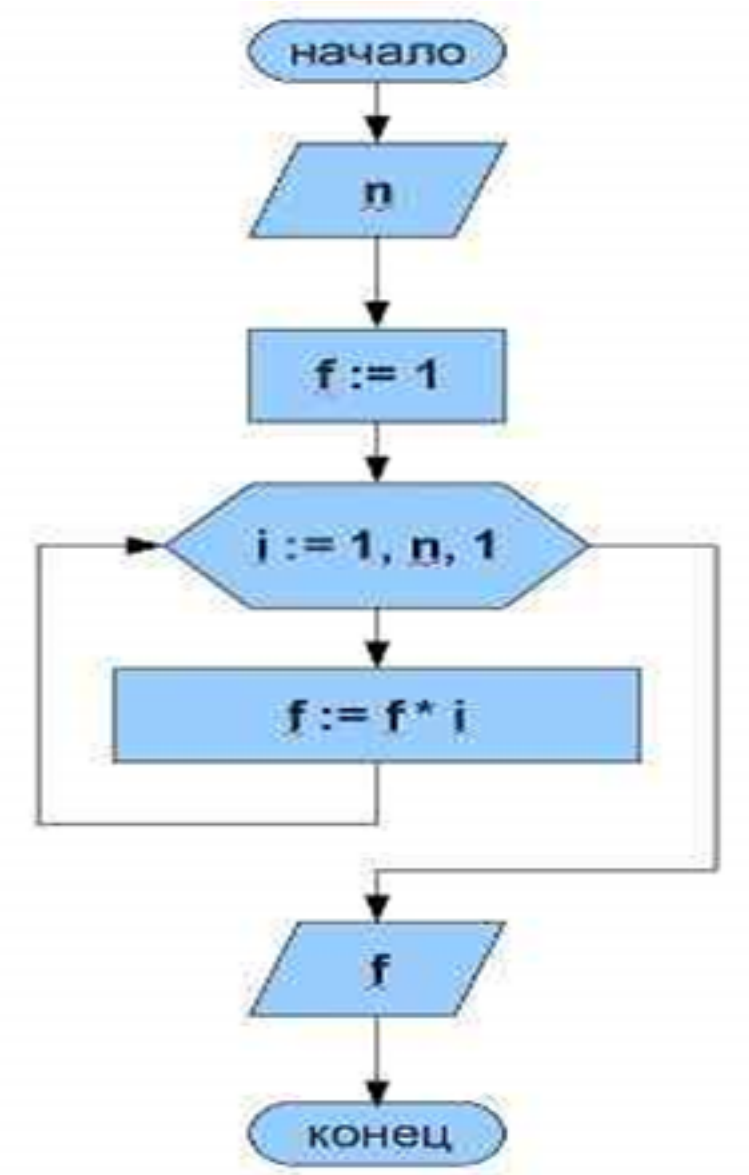
12. На картинке представлена блок-схема. Выберите нужный оператор для построения кода. (1 балл)

" while"

" repeat"

" for"

" joy"



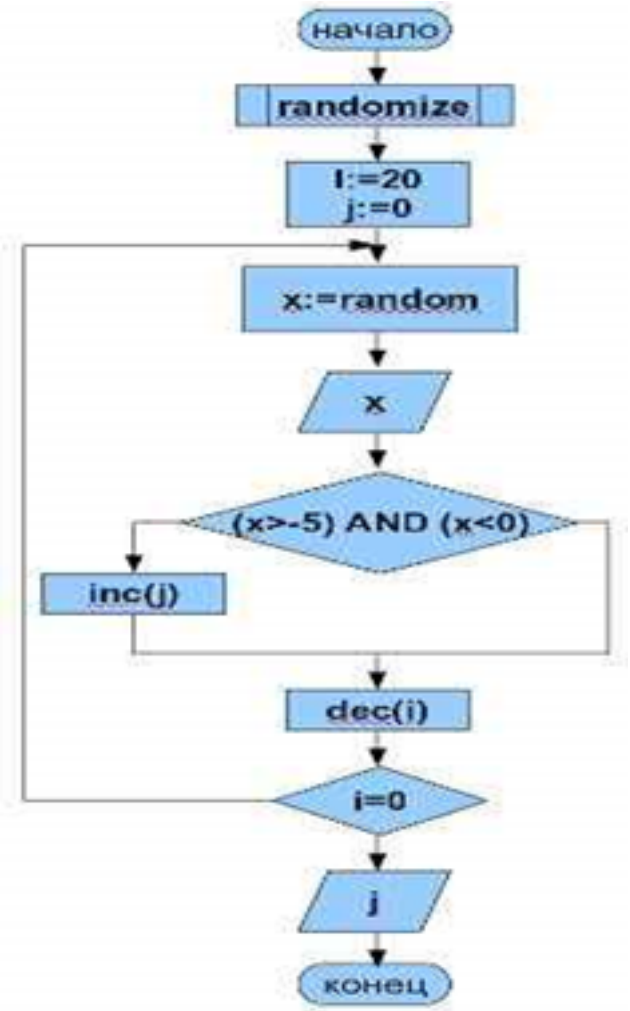
13. На картинке представлена блок-схема. Выберите нужный оператор для построения кода. (1 балл)

    " while"

    " repeat"

    " for"

    " joy"



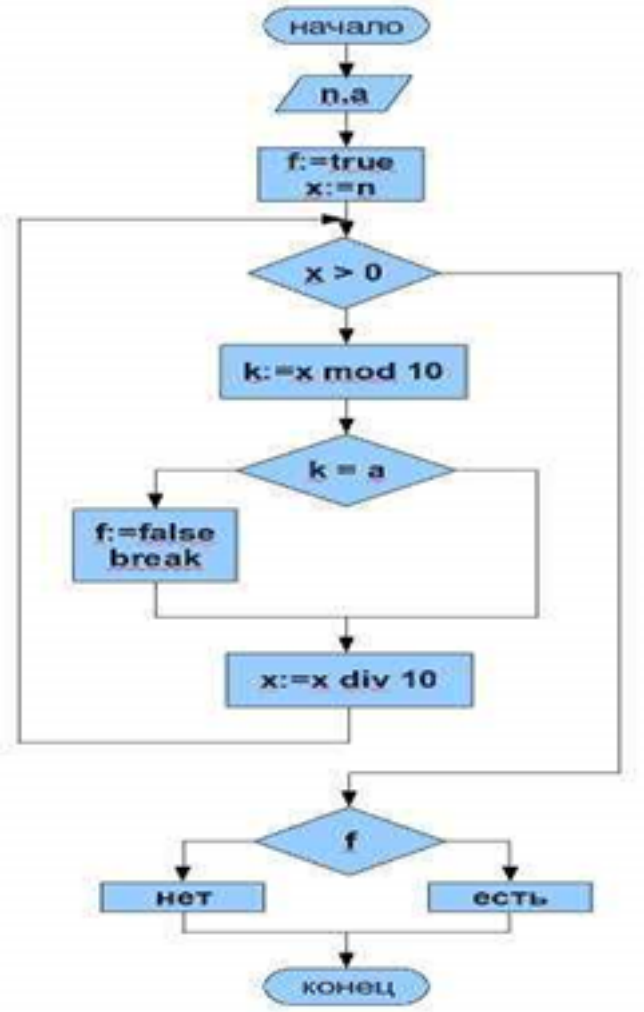
14. На картинке представлена блок-схема. Выберите нужный оператор для построения кода. (1 балл)

    " while"

    " repeat"

    " for"

    " joy"



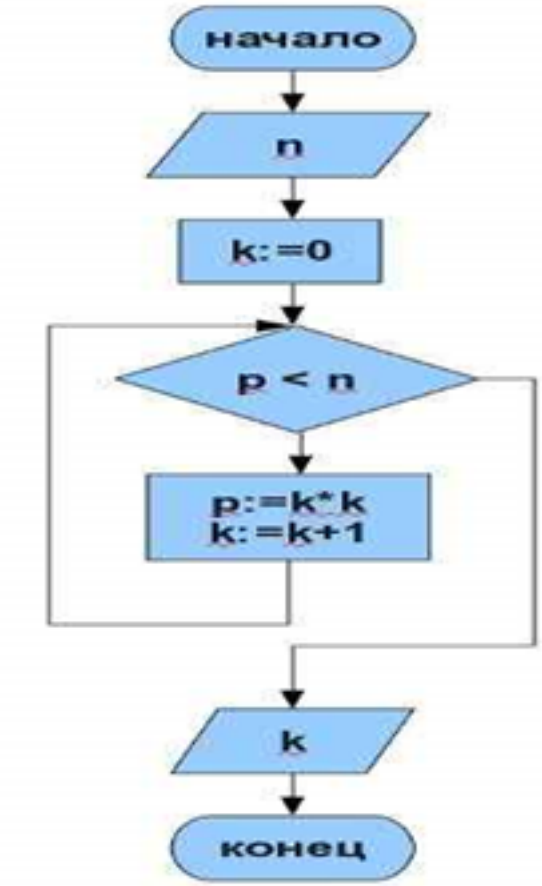
15. На картинке представлена блок-схема. Выберите нужный оператор для построения кода. (1 балл)

    " while"

    " repeat"

    " for"

    " joy"



16. Выберите верное утверждение (1 балл)

    Переменная состоит из имени и выделенной области памяти, которая ему соответствует

    Числа являются частью переменных

    После присваивания переменная запоминает число

    Переменные в области памяти задаются независимо от средств вычисления и необходимостей автора. Невозможно изменить нахождения переменных в области памяти

17. Выберите верное утверждение (1 балл):

    При нахождением max и min перед циклом необходимо присвоить max ноль, min – максимальное число

    При нахождением max и min перед циклом необходимо присвоить max максимальное число, min – ноль

    При нахождением max и min перед циклом необходимо присвоить max, min – максимальное число

    При нахождением max и min перед циклом необходимо присвоить max, min – ноль

Оценка	Показатели оценки
3	Обучающийся выполнил 6 - 15 заданий
4	Обучающийся выполнил 16 - 19 заданий
5	Обучающийся выполнил 20 - 22 задания

Задание №3

Описать методы вычисления сложности работы алгоритмов.

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 один из двух методов: 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.
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

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.