

Документ подписан простой электронной подписью
 Информация о владельце:
 ФИО: ЧУМАЧЕНКО ТАТЬЯНА АЛЕКСАНДРОВНА
 Должность: РЕКТОР
 Дата подписания: 21.10.2022 15:38:31
 Уникальный программный ключ:
 9c9f7aaffa4840d284abe156657b8f85432bdb16




МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
ГУМАНИТАРНО-ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГГПУ»)

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
(ОЦЕНОЧНЫЕ СРЕДСТВА)



Шифр	Наименование дисциплины (модуля)
ФТД	Методика решения задач повышенной сложности по информатике

Код направления подготовки	44.04.01
Направление подготовки	Педагогическое образование
Наименование (я) ОПОП (направленность / профиль)	Информатика и робототехника в образовании
Уровень образования	магистр
Форма обучения	заочная

Разработчики:

Должность	Учёная степень, звание	Подпись	ФИО
Старший преподаватель			Боровская Елена Владимировна

Рабочая программа рассмотрена и одобрена (обновлена) на заседании кафедры (структурного подразделения)

Кафедра	Заведующий кафедрой	Номер протокола	Дата протокола	Подпись
Кафедра информатики, информационных технологий и методики обучения информатике	Рузаков Андрей Александрович	10	13.06.2019	
Кафедра информатики, информационных технологий и методики обучения информатике	Рузаков Андрей Александрович	1	10.09.2020	

Раздел 1. Компетенции обучающегося, формируемые в результате освоения образовательной программы с указанием этапов их формирования

Таблица 1 - Перечень компетенций, с указанием образовательных результатов в процессе освоения дисциплины (в соответствии с РПД)

Формируемые компетенции			
Индикаторы ее достижения	Планируемые образовательные результаты по дисциплине		
	знать	уметь	владеть

ПК-3 способен организовать образовательный процесс в системе общего, профессионального и дополнительного образования

ПК.3.1 Знает основы педагогического мастерства	3.1 иметь представление о требованиях к уровню подготовки выпускников, достижение которого проверяется на едином государственном экзамене по информатике		
ПК.3.2 Умеет организовать образовательный процесс в системе общего, профессионального и дополнительного образования		У.1 Умеет организовать процесс подготовки к ЕГЭ в рамках элективных курсов	
ПК.3.3 Владеет методами организации образовательного процесса в системе общего, профессионального и дополнительного образования			В.1 владеть методами решения задач повышенной сложности ЕГЭ по информатике

УК-6 способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки

УК-6.1 Знает психолого-педагогические основы самооценки, саморазвития, самореализации; направления и источники саморазвития и самореализации; способы самоорганизации собственной деятельности и ее совершенствования	3.2 элементы содержания, проверяемые на ГИА и ЕГЭ по информатике;		
УК-6.2 Умеет определять приоритеты собственной деятельности и прогнозировать пути ее совершенствования; осуществлять контроль, оценку и рефлексию собственной деятельности на основе личностных и профессиональных приоритетов		У.2 уметь применять методику обучения информатике для обучения школьников в рамках элективных курсов решению задач повышенной сложности, включаемых в ГИА и ЕГЭ по информатике;	

УК-6.3 Владеет навыками осуществления деятельности по самоорганизации и саморазвитию в соответствии с личностными и профессиональными приоритетами			В.2 обладать практическими навыками решения типовых задач повышенной сложности.
--	--	--	---

Компетенции связаны с дисциплинами и практиками через матрицу компетенций согласно таблице 2.

Таблица 2 - Компетенции, формируемые в результате обучения

Код и наименование компетенции	
Составляющая учебного плана (дисциплины, практики, участвующие в формировании компетенции)	Вес дисциплины в формировании компетенции (100 / количество дисциплин, практик)
ПК-3 способен организовать образовательный процесс в системе общего, профессионального и дополнительного образования	
Дифференцированное обучение информатике	33,33
Подготовка к итоговой аттестации учащихся старшей профильной школы	33,33
Основы научно-технического творчества	33,33
УК-6 способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	
Методология и методы психолого-педагогического исследования	14,29
Инновационные процессы в образовании	14,29
производственная практика (научно-исследовательская работа)	14,29
производственная практика (педагогическая)	14,29
Экзамен по модулю "Методология исследования в образовании"	14,29
Цифровая образовательная среда школы	14,29
учебная практика (по информатике и робототехнике)	14,29

Таблица 3 - Этапы формирования компетенций в процессе освоения ОПОП

Код компетенции	Этап базовой подготовки	Этап расширения и углубления подготовки	Этап профессионально-практической подготовки
ПК-3	Дифференцированное обучение информатике, Подготовка к итоговой аттестации учащихся старшей профильной школы, Основы научно-технического творчества		
УК-6	Методология и методы психолого-педагогического исследования, Инновационные процессы в образовании, производственная практика (научно-исследовательская работа), производственная практика (педагогическая), Экзамен по модулю "Методология исследования в образовании", Цифровая образовательная среда школы, учебная практика (по информатике и робототехнике)		производственная практика (научно-исследовательская работа), производственная практика (педагогическая), учебная практика (по информатике и робототехнике)

Раздел 2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Таблица 4 - Показатели оценивания компетенций на различных этапах их формирования в процессе освоения учебной дисциплины (в соответствии с РПД)

№	Раздел
Формируемые компетенции	
Показатели сформированности (в терминах «знать», «уметь», «владеть»)	
1	Решение задач школьного курса информатики
ПК-3 УК-6	
Знать иметь представление о требованиях к уровню подготовки выпускников, достижение которого проверяется на едином государственном экзамене по информатике Знать элементы содержания, проверяемые на ГИА и ЕГЭ по информатике;	Задача
Уметь умеет организовать процесс подготовки к ЕГЭ в рамках элективных курсов Уметь уметь применять методику обучения информатике для обучения школьников в рамках элективных курсов решению задач повышенной сложности, включаемых в ГИА и ЕГЭ по информатике;	Задача
Владеть владеть методами решения задач повышенной сложности ЕГЭ по информатике Владеть обладать практическими навыками решения типовых задач повышенной сложности.	Задача

Таблица 5 - Описание уровней и критериев оценивания компетенций, описание шкал оценивания

Код	Содержание компетенции			
Уровни освоения компетенции	Содержательное описание уровня	Основные признаки выделения уровня (критерии оценки сформированности)	Пятибалльная шкала (академическая оценка)	% освоения (рейтинговая оценка)
ПК-3	ПК-3 способен организовать образовательный процесс в системе общего, профессионального и дополнительного образования			
УК-6	УК-6 способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки			

Раздел 3. Типовые контрольные задания и (или) иные материалы, необходимые для оценки планируемых результатов обучения по дисциплине (модулю)

1. Оценочные средства для текущего контроля

Раздел: Решение задач школьного курса информатики

Задания для оценки знаний

1. Задача:

На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N .

2. К этой записи дописываются справа ещё два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия – справа дописывается остаток от деления суммы цифр на 2. Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R . Укажите такое наименьшее число R , которое превышает 118 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе счисления.

Определите, количество чисел k , для которых следующая программа выведет такой же результат, что и для $k = 13$?

Паскаль

```
var k, i : longint;  
function f(n: longint): longint;  
begin  
  f:= n*n*n - 5*n;  
end;  
begin  
  readln(k);  
  i := 1;  
  while f(i) < k do  
    i := i + 1;  
  if 2*f(i)-k <= k-f(i-1) then  
    writeln(i)  
  else writeln(i-1);  
end.
```

Требовалось написать программу, которая получает на вход натуральное число N , не превосходящее 109, и выводит число, которое получается из N после удаления всех единиц; порядок остальных цифр при этом не меняется. Например, число 19520125 должно быть преобразовано в число 952025. Число, в котором все цифры – единицы и нули, должно быть преобразовано в 0. Незначащие нули в старших разрядах полученного числа печатать не нужно. Программист написал программу неправильно.

Паскаль

```
var N, R, T: longint;  
d: integer;  
begin  
  readln(N);  
  R:=0;  
  T:=1;  
  while N>0 do begin  
    d := N mod 10;  
    if d<>1 then begin  
      R := R + d*T;  
      T := T+1  
    end;  
    N := N div 10;  
  end;  
  writeln(T);  
end.
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 314.

2. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

- 1) выпишите строку, в которой сделана ошибка;
- 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Дан массив, содержащий 70 неотрицательных целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшую нечётную сумму двух соседних элементов массива. Гарантируется, что в массиве есть соседние элементы с нечётной суммой.

Паскаль

```
const n = 70;
var
  a: array [0..n-1]
    of integer;
  i, j, x, y: integer;
begin
  for i := 0 to n-1 do
    readln(a[i]);
  ...
end.
```

Дан список точек плоскости с целочисленными координатами. Необходимо определить:

- 1) номер координатной четверти K , в которой находится больше всего точек;
- 2) точку A в этой четверти, наименее удалённую от осей координат;
- 3) расстояние R от этой точки до ближайшей оси.

Если в нескольких четвертях расположено одинаковое количество точек, следует выбрать ту четверть, в которой величина R меньше. При равенстве и количества точек, и величины R необходимо выбрать четверть с меньшим номером K . Если в выбранной четверти несколько точек находятся на одинаковом минимальном расстоянии от осей координат, нужно выбрать первую по списку. Точки, хотя бы одна из координат которых равна нулю, считаются не принадлежащими ни одной четверти и не рассматриваются. Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу.

В первой строке вводится одно целое положительное число – количество точек N . Каждая из следующих N строк содержит координаты очередной точки – два целых числа (первое – координата x , второе – координата y).

Пример входных данных:

```
7
-3 4
1 2
1 1
0 4
-2 -3
-6 8
-12 1
```

Программа должна вывести номер выбранной четверти K , количество точек в ней M , координаты выбранной точки A и минимальное расстояние R по образцу, приведённому ниже в примере.

Пример выходных данных для приведённого выше примера входных данных:

```
K = 2
M = 3
A = (-12, 1)
R = 1
```

Задания для оценки умений

1. Задача:

На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N .
2. К этой записи дописываются справа ещё два разряда по следующему правилу:
 - а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия – справа дописывается остаток от деления суммы цифр на 2. Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R. Укажите такое наименьшее число R, которое превышает 118 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе счисления.

Определите, количество чисел k, для которых следующая программа выведет такой же результат, что и для k = 13?

Паскаль

```
var k, i : longint;
function f(n: longint): longint;
begin
  f:= n*n*n - 5*n;
end;
begin
  readln(k);
  i := 1;
  while f(i) < k do
    i := i + 1;
  if 2*f(i)-k <= k-f(i-1) then
    writeln(i)
  else writeln(i-1);
end.
```

Требовалось написать программу, которая получает на вход натуральное число N, не превосходящее 109, и выводит число, которое получается из N после удаления всех единиц; порядок остальных цифр при этом не меняется. Например, число 19520125 должно быть преобразовано в число 952025. Число, в котором все цифры – единицы и нули, должно быть преобразовано в 0. Незначащие нули в старших разрядах полученного числа печатать не нужно. Программист написал программу неправильно.

Паскаль

```
var N, R, T: longint;
    d: integer;
begin
  readln(N);
  R:=0;
  T:=1;
  while N>0 do begin
    d := N mod 10;
    if d<>1 then begin
      R := R + d*T;
      T := T+1
    end;
    N := N div 10;
  end;
  writeln(T);
end.
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 314.

2. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

- 1) выпишите строку, в которой сделана ошибка;
- 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Дан массив, содержащий 70 неотрицательных целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшую нечётную сумму двух соседних элементов массива. Гарантируется, что в массиве есть соседние элементы с нечётной суммой.

Паскаль

```
const n = 70;
var
  a: array [0..n-1]
    of integer;
  i, j, x, y: integer;
begin
  for i := 0 to n-1 do
    readln(a[i]);
  ...
```

end.

Дан список точек плоскости с целочисленными координатами. Необходимо определить:

- 1) номер координатной четверти K , в которой находится больше всего точек;
- 2) точку A в этой четверти, наименее удалённую от осей координат;
- 3) расстояние R от этой точки до ближайшей оси.

Если в нескольких четвертях расположено одинаковое количество точек, следует выбрать ту четверть, в которой величина R меньше. При равенстве и количества точек, и величины R необходимо выбрать четверть с меньшим номером K . Если в выбранной четверти несколько точек находятся на одинаковом минимальном расстоянии от осей координат, нужно выбрать первую по списку. Точки, хотя бы одна из координат которых равна нулю, считаются не принадлежащими ни одной четверти и не рассматриваются. Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу.

В первой строке вводится одно целое положительное число – количество точек N . Каждая из следующих N строк содержит координаты очередной точки – два целых числа (первое – координата x , второе – координата y).

Пример входных данных:

```
7
-3 4
1 2
1 1
0 4
-2 -3
-6 8
-12 1
```

Программа должна вывести номер выбранной четверти K , количество точек в ней M , координаты выбранной точки A и минимальное расстояние R по образцу, приведённому ниже в примере.

Пример выходных данных для приведённого выше примера входных данных:

```
K = 2
M = 3
A = (-12, 1)
R = 1
```

Задания для оценки владений

1. Задача:

На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N .

2. К этой записи дописываются справа ещё два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия – справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R . Укажите такое наименьшее число R , которое превышает 118 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе счисления.

Определите, количество чисел k , для которых следующая программа выведет такой же результат, что и для $k = 13$?

Паскаль

```
var k, i : longint;
function f(n: longint): longint;
begin+
  f:= n*n*n - 5*n;
end;
begin
  readln(k);
  i := 1;
  while f(i) < k do
    i := i + 1;
  if 2*f(i)-k <= k-f(i-1) then
```



```

        writeln(i)
    else writeln(i-1);
end.

```

Требовалось написать программу, которая получает на вход натуральное число N , не превосходящее 109, и выводит число, которое получается из N после удаления всех единиц; порядок остальных цифр при этом не меняется. Например, число 19520125 должно быть преобразовано в число 952025. Число, в котором все цифры – единицы и нули, должно быть преобразовано в 0. Незначащие нули в старших разрядах полученного числа печатать не нужно. Программист написал программу неправильно.

Паскаль

```

var N, R, T: longint;
    d: integer;
begin
    readln(N);
    R:=0;
    T:=1;
    while N>0 do begin
        d := N mod 10;
        if d<>1 then begin
            R := R + d*T;
            T := T+1
        end;
        N := N div 10;
    end;
    writeln(T);
end.

```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 314.
2. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:
 - 1) выпишите строку, в которой сделана ошибка;
 - 2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Дан массив, содержащий 70 неотрицательных целых чисел. Опишите на одном из языков программирования алгоритм, позволяющий найти и вывести наименьшую нечётную сумму двух соседних элементов массива. Гарантируется, что в массиве есть соседние элементы с нечётной суммой.

Паскаль

```

const n = 70;
var
    a: array [0..n-1]
        of integer;
    i, j, x, y: integer;
begin
    for i := 0 to n-1 do
        readln(a[i]);
    ...
end.

```

Дан список точек плоскости с целочисленными координатами. Необходимо определить:

- 1) номер координатной четверти K , в которой находится больше всего точек;
- 2) точку A в этой четверти, наименее удалённую от осей координат;
- 3) расстояние R от этой точки до ближайшей оси.

Если в нескольких четвертях расположено одинаковое количество точек, следует выбрать ту четверть, в которой величина R меньше. При равенстве и количества точек, и величины R необходимо выбрать четверть с меньшим номером K . Если в выбранной четверти несколько точек находятся на одинаковом минимальном расстоянии от осей координат, нужно выбрать первую по списку. Точки, хотя бы одна из координат которых равна нулю, считаются не принадлежащими ни одной четверти и не рассматриваются. Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу.

В первой строке вводится одно целое положительное число – количество точек N . Каждая из следующих N строк содержит координаты очередной точки – два целых числа (первое – координата x , второе – координата y).

Пример входных данных:

7

-3 4
1 2
1 1
0 4
-2 -3
-6 8
-12 1

Программа должна вывести номер выбранной четверти K, количество точек в ней M, координаты выбранной точки A и минимальное расстояние R по образцу, приведённому ниже в примере.

Пример выходных данных для приведённого выше примера входных данных:

K = 2
M = 3
A = (-12, 1)
R = 1

2. Оценочные средства для промежуточной аттестации

1. Зачет по факультативу

Вопросы к зачету:

1. Спецификация контрольных измерительных материалов единого государственного экзамена и ГИА
2. кодификатор элементов содержания и требований к уровню подготовки выпускников
3. статистика сдачи ГИА и ЕГЭ по информатике,
4. Нахождение наибольшего общего делителя двух натуральных чисел (алгоритм Евклида).
5. Использование цикла для решения простых переборных задач (поиск наименьшего простого делителя данного натурального числа, проверка числа на простоту, и т.д.)
6. Запись натурального числа в позиционной системе с основанием меньшим или равным 10.
7. Обработка и преобразование такой записи числа.
8. Нахождение минимума и максимума двух, трех, четырех данных чисел без использования массивов и циклов.
9. Нахождение всех корней заданного квадратного уравнения.
10. Операции с элементами массива. Линейный поиск элемента.
11. Вставка и удаление элементов в массиве.
12. Перестановка элементов данного массива в обратном порядке.
13. Суммирование элементов массива.
14. Проверка соответствия элементов массива некоторому условию
15. Нахождение сумм, произведений элементов данной конечной числовой последовательности (или массива).
16. Заполнение элементов одномерного и двумерного массива по заданным правилам.
17. Нахождение минимального (максимального) значения в данном массиве и количества элементов, равных ему, за однократный просмотр массива.
18. Нахождение второго по величине второго максимального или второго минимального) значения в данном массиве за однократный просмотр массива.
19. Операции с элементами массива, отобранными по некоторому условию (например, нахождение минимального четного элемента в массиве, нахождение количества и суммы всех четных элементов в массиве)
20. Сортировка массива.
21. Слияние двух упорядоченных массивов в один без использования сортировки
22. Обработка отдельных символов данной строки.
23. Подсчет частоты появления символа в строке.
24. Работа с подстроками данной строки с разбиением на слова по пробельным символам.
25. Поиск подстроки внутри данной строки, замена найденной подстроки на другую строку

Раздел 4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

1. Для текущего контроля используются следующие оценочные средства:

1. Задача

Задачи позволяют оценивать и диагностировать знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, узнавание объектов изучения в рамках определенного раздела дисциплины;

умения синтезировать, анализировать, обобщать фактический и теоретический материал с формулированием конкретных выводов, установлением причинно-следственных связей.

Алгоритм решения задач:

1. Внимательно прочитайте условие задания и уясните основной вопрос, представьте процессы и явления, описанные в условии.
2. Повторно прочтите условие для того, чтобы чётко представить основной вопрос, проблему, цель решения, заданные величины, опираясь на которые можно вести поиск решения.
3. Произведите краткую запись условия задания.
4. Если необходимо, составьте таблицу, схему, рисунок или чертёж.
5. Установите связь между искомыми величинами и данными; определите метод решения задания, составьте план решения.
6. Выполните план решения, обосновывая каждое действие.
7. Проверьте правильность решения задания.
8. Произведите оценку реальности полученного решения.
9. Запишите ответ.

2. Описание процедуры промежуточной аттестации

Оценка за зачет/экзамен может быть выставлена по результатам текущего рейтинга. Текущий рейтинг – это результаты выполнения практических работ в ходе обучения, контрольных работ, выполнения заданий к лекциям (при наличии) и др. видов заданий.

Результаты текущего рейтинга доводятся до студентов до начала экзаменационной сессии.

Цель зачета – проверка и оценка уровня полученных студентом специальных знаний по факультативу и соответствующих им умений и навыков, а также умения логически мыслить, аргументировать избранную научную позицию, реагировать на дополнительные вопросы, ориентироваться в массиве информации. Подготовка к зачету начинается с первого занятия по факультативу, на котором обучающиеся получают предварительный перечень вопросов к зачёту и список рекомендуемой литературы, их ставят в известность относительно критериев выставления зачёта и специфике текущей и итоговой аттестации. С самого начала желательно планомерно осваивать материал, руководствуясь перечнем вопросов к зачету и списком рекомендуемой литературы, а также путём самостоятельного конспектирования материалов занятий и результатов самостоятельного изучения учебных вопросов.

По результатам сдачи зачета выставляется оценка «зачтено» или «не зачтено».