Управление культуры Администрации города Екатеринбурга
Муниципальное бюджетное образовательное учреждение
высшего образования
«Екатеринбургская академия современного искусства» (институт)

УТВЕРЖДАЮ
Ректор
_____И.А. Ахьямова
«01» апреля 2023 г.

ПРОГРАММА
вступительного испытания по дисциплине
ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ
для поступающих на базе среднего профессионального образования

Направление подготовки
09.03.03 Прикладная информатика

Екатеринбург
2023

Программа вступительного испытания для поступающих в МБОУ ВО ЕАСИ на базе среднего профессионального образования по дисциплине **ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ** для направления подготовки бакалавров 09.03.03 Прикладная информатика

Составители:к.п.н., доцент кафедры информационных систем и технологий ФГАОУ ВО «РГППУ» И.А. Суслова, к.п.н., доцент кафедры прикладной информатики МБОУ ВО ЕАСИ Е.Б. Стариченко

СОДЕРЖАНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА 4
СТРУКТУРА ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ 4
СОДЕРЖАНИЕ РАЗДЕЛОВ ПРОГРАММЫ 5
СПИСОК ЛИТЕРАТУРЫ, РЕКОМЕНДУЕМОЙ ДЛЯ ПОДГОТОВКИ 8
ПРИМЕР ЭКЗАМЕНАЦИОННОГО ЗАДАНИЯ 9
ИНСТРУКЦИЯ ПО ПРОВЕРКЕ И ОЦЕНКЕ РАБОТ 24

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа предназначена для абитуриентов, поступающих на базе среднего профессионального образования по направлению подготовки бакалавров 09.03.03 Прикладная информатика на основании результатов вступительных испытаний, проводимых вузом.

Категории поступающих имеющих право поступать на обучение по результатам общеобразовательных вступительных испытаний, проводимых институтом самостоятельно определяются в соотвтестии с Правилами приема граждан на обучение по программам бакалавриата в 2022 году в МБОУ ВО «Екатеринбургская академия современного искусства» (институт) (далее Правила приема).

Максимальная продолжительность вступительного испытания – 3 часа 55 минут (235 мин), допускается досрочное завершение тестирования по желанию испытуемого.

Объявление результата вступительного испытания, ознакомление с экзаменационной работой, прием и рассмотрение апелляций производится в соответствии с Правилами приема.

СТРУКТУРА ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ

В экзаменационной работе содержатся задания по 10 темам курса «Информатика и ИКТ», составляющим основное содержание предмета.

№	Название раздела	Максимальный балл за задания данного вида
1	Информация и её кодирование	**12,5**
2	Моделирование и компьютерный эксперимент	**2,5**
3	Системы счисления	**7,5**
4	Основы логики	**12,5**
5	Элементы теории алгоритмов	**27,5**
6	Программные средства информационных и коммуникационных технологий	**2,5**
7	Технология обработки графической и звуковой информации	**2,5**
8	Технология обработки текстовой и числовой информации	**5**
9	Технология хранения, поиска и сортировки в базах данных	**5**
10	Основные устройства информационных и коммуникационных технологий	**2,5**
11	Алгоритмизация и программирование	**20**
	Итого:	**100**

Экзаменационная работа состоит из 2 частей, включающих 27 заданий. На выполнение части 1 работы рекомендуется отводить 1,5 часа (90 минут). На выполнение заданий части 2 – 2,5 часа (150 минут).

Экзаменационная работа состоит из двух частей, включающих в себя 27 заданий.

Часть 1 содержит 23 задания с кратким ответом.

Часть 2 содержит 4 задания с развёрнутым ответом.

Итоговой оценкой за вступительное испытание является сумма баллов за все задания.

Максимальное количество баллов, которое может получить абитуриент по итогам вступительного испытания, равняется ***100 баллам.***

Минимальное количество баллов, подтверждающее успешное прохождение вступительного испытания равняется ***42 баллам.***

Абитуриенты, получившие на вступительном испытании результат ниже установленного минимального количества баллов выбывают из конкурса по данному направлению подготовки.

СОДЕРЖАНИЕ РАЗДЕЛОВ ПРОГРАММЫ

1. Информация и ее кодирование

Информация и информационные процессы в технике. Кодирование информации с помощью знаковых систем. Знаки: форма и значение. Знаковые системы. Кодирование информации. Количество информации. Количество информации как мера уменьшения неопределенности знания. Определение количества информации. Алфавитный подход к определению количества информации.

2. Основы логики

Основные понятия алгебры логики. Понятие высказывания. Логические выражения и логические операции: НЕ, ИЛИ, И, ЕСЛИ... ,ТО..., эквивалентность. Таблицы истинности.

Составление таблиц истинности по логической формуле. Законы булевой алгебры. Определение логического выражения по таблице истинности.

Логические элементы и основные логические устройства компьютера.

3. Моделирование и компьютерный эксперимент

Моделирование как метод познания. Системный подход в моделировании. Формы представления моделей. Формализация. Основные этапы разработки и исследования моделей на компьютере. Исследование интерактивных компьютерных моделей. Исследование физических моделей. Исследование астрономических моделей. Исследование алгебраических моделей. Исследование геометрических моделей (планиметрия). Исследование геометрических моделей

(стереометрия). Исследование химических моделей. Исследование биологических моделей.

4. Информационная безопасность

Защита от несанкционированного доступа к информации. Защита с использованием паролей. Биометрические системы защиты. Физическая защита данных на дисках. Защита от вредоносных программ. Вредоносные и антивирусные программы. Компьютерные вирусы и защита от них. Сетевые черви и защита от них. Троянские программы и защита от них. Хакерские утилиты и защита от них.

5. Основные устройства информационных и коммуникационных технологий

История развития вычислительной техники. Архитектура персонального компьютера. Операционные системы. Основные характеристики операционных систем. Операционная система Linux. Локальные и глобальные компьютерные сети, организации компьютерных сетей. Аппаратные средства построения сети.

6. Программные средства информационных и коммуникационных технологий

Возможности Интернета. Среда браузера Internet Explorer. Поиск информации в сети Интернет. Язык разметки гипертекста HTML. Веб-страница с графическими объектами. Веб-страница с гиперссылками. Мир электронной почты.

7. Технология обработки текстовой и числовой информации

Макет текстового документа. Характеристика текстового процессора. Объекты текстового документа и их параметры. Способы выделения объектов текстового документа.

Создание и редактирование документа в среде текстового процессора. Форматирование текста. Оформление текста в виде таблицы и печать документа. Использование в текстовом документе графических объектов.

Назначение табличного процессора. Объекты документа табличного процессора. Данные электронной таблицы. Типовые действия над объектами электронной таблицы.

Создание и редактирование документа в среде табличного документа. Форматирование табличного документа.

Правила записи формул и функций. Копирование формул в табличном документе. Использование функций и логических формул в табличном документе.

Представление данных в виде диаграмм в среде табличного документа.

8. Технология хранения, поиска и сортировки в базах данных

Табличные базы данных. Система управления базами данных. Основные объекты СУБД: таблицы, формы, запросы, отчеты. Использование формы для просмотра и редактирования записей в табличной базе данных. Поиск записей в табличной базе данных с помощью фильтров и запросов. Сортировка записей в табличной базе данных. Печать данных с помощью отчетов. Иерархические базы данных. Сетевые базы данных.

9. Технология обработки графической и звуковой информации

Назначение графических редакторов. Растровая и векторная графика. Объекты растрового редактора. Типовые действия над объектами. Инструменты графического редактора.

Создание и редактирование рисунка в среде графического редактора. Создание и редактирование рисунка с текстом.

Выполнение геометрических построений в системе компьютерного черчения

Создание и редактирование оцифрованного звука

Разработка мультимедийной интерактивной презентации

10. Алгоритмизация и программирование

Программирование в среде на выбор экзаменующегося: инструментарий среды; информационная модель объекта; программы для реализации типовых конструкций алгоритмов (последовательного, циклического, разветвляющегося); понятия процедуры и модуля; процедура с параметрами; функции; инструменты логики при разработке программ, моделирование системы.

СПИСОК ЛИТЕРАТУРЫ, РЕКОМЕНДУЕМОЙ ДЛЯ ПОДГОТОВКИ

1. *Глизбург В.И.* ЕГЭ. Информатика и ИКТ. Комплексная подготовка / В.И. Глизбург, Е.С. Самойлова. – М.: Айрис-Пресс, 2013. – 336 с.
2. *Зейдельман Я.Н.* Информатика: Подготовка к ЕГЭ в 2013 году: диагностические работы / Я.Н. Зейдельман, М.Я. Ройтберг. – М.: Московский центр непрерывного математического образования (МЦНМО), 2013. – 188 с.
3. *Зорина Е.М.* ЕГЭ-2013. Информатика: сборник заданий / Е.М. Зорина, М.В. Зорин. – М.: Эксмо-Пресс, 2012. – 224 с.
4. Информатика и ИКТ: Подготовка к ЕГЭ 2011: типовые задачи / под ред. Н.В. Макаровой. – СПб. : Питер, 2011. – 464 с.
5. *Лещинер В.Р.* ЕГЭ 2013. Информатика. Оптимальный банк заданий для подготовки учащихся / В.Р. Лещинер. – М.: Экзамен, 2013. – 136 с.
6. *Лысенко Ф.Ф.* Подготовка к ЕГЭ-2013: Информатика и ИКТ / Ф.Ф. Лысенко, Л.Н. Евич. – М.: Легион, 2013. – 432 с.
7. *Макарова Н.В.* Информатика и ИКТ : 11 класс : базовый уровень / Н.В. Макарова, Г.С. Николайчук, Ю.Ф. Титова – СПб. : Питер. – 224 с.
8. *Трофимов В.В.* Информатика: учебник / под ред. проф. В.В. Трофимова. – М.: ИД Юрайт, 2013. – 917 с.
9. *Трофимова Л.А.* Методы принятия управленческих решений: учебник для бакалавров / Л.А.Трофимова, В.В. Трофимов. – М.: Юрайт. 2013. – 335 с.
10. *Угринович Н.Д.* Информатика и ИКТ: 11 класс / Н.Д. Угринович. – М.: Бином. Лаборатория знаний, 2011. – 192 с.
11. *Ушаков Д.* ЕГЭ-2013. Информатика: Самое полное издание типовых вариантов заданий / Д. Ушаков, А. Якушкин. М.: Астрель, 2013. – 318 с.

ПРИМЕР ЭКЗАМЕНАЦИОННОГО ЗАДАНИЯ

В экзаменационных заданиях используются следующие соглашения:

1. Обозначения для логических связок (операций):
   a) *отрицание* (инверсия, логическое НЕ) обозначается ¬ (например, ¬ A);
   b) *конъюнкция* (логическое умножение, логическое И) обозначается /\ (например, A /\ B) либо & (например, A & B);
   c) *дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается \/ (например, A \/ B) либо | (например, A | B);
   d) *следование* (импликация) обозначается -> (например, A -> B);
   e) символ 1 используется для обозначения истины (истинного высказывания); символ 0 - для обозначения лжи (ложного высказывания).

2. Два логических выражения, содержащих переменные, называются *равносильными* (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения A -> B и (¬ A) \/ B равносильны, а A \/ B и A /\ B - нет (значения выражений разные, например, при A=1, B=0).

3. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), эквивалентность (равносильность). Таким образом, ¬ A /\ B \/ C/\ D совпадает с ((¬ A)/\ B) \/ (C/\ D). Возможна запись A /\ B /\ C вместо (A /\ B) /\ C. То же относится и к дизъюнкции: возможна запись A \/ B \/ C вместо (A \/ B) \/ C.

Часть 1

Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в строке под заданием.

1 Сколько единиц в двоичной записи восьмеричного числа 1731_8? *Ответ запишите в строке под заданием*

---

2 Логическая функция F задаётся выражением $(x \vee y) \rightarrow (y \equiv z)$. На рисунке приведён частично заполненный фрагмент таблицы истинности функции F, содержащий неповторяющиеся строки. Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z.

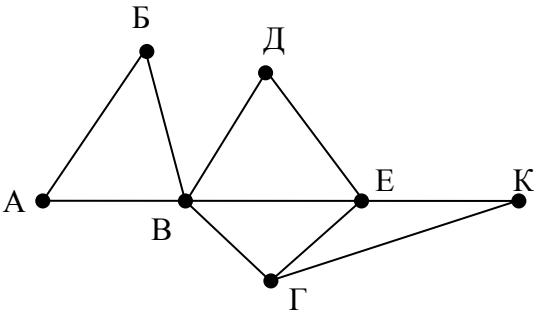
?	?	?	F
0	0		0
0			0

В ответе напишите буквы x, y, z в том порядке, в котором идут соответствующие им столбцы. Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно. *Ответ запишите в строке под заданием*

---

3 На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содер-жатся сведения о длинах этих дорог (в километрах). Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта В в пункт Е. В ответе запишите целое число – так, как оно указано в таблице. *Ответ запишите в строке под заданием.*

	П1	П2	П3	П4	П5	П6	П7
П1		45		10			
П2	45			40		55	
П3					15	60	
П4	10	40				20	35
П5			15			55	
П6		55	60	20	55		45
П7				35		45	



---

4 В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите, сколько прямых потомков (т.е. детей и внуков) Павленко А.К. упомянуты в таблице 1. *Ответ запишите в строке под заданием.*

Таблица 1

ID	Фамилия_И.О.	Пол
2146	Кривич Л.П.	Ж
2155	Павленко А.К.	М
2431	Хитрук П.А.	М
2480	Кривич А.А.	М
2302	Павленко Е.А.	Ж
2500	Сокол Н.А.	Ж
3002	Павленко И.А.	М
2523	Павленко Т.Х.	Ж
2529	Хитрук А.П	М
2570	Павленко П.И.	М
2586	Павленко Т.И.	Ж
2933	Симонян А.А.	Ж
2511	Сокол В.А.	Ж
3193	Биба С.А.	Ж

Таблица 2

ID_Родителя	ID_Ребенка
2146	2302
2146	3002
2155	2302
2155	3002
2302	2431
2302	2511
2302	3193
3002	2586
3002	2570
2523	2586
2523	2570
2529	2431
2529	2511

---

5 По каналу связи с помощью равномерного двоичного кода передаются сообщения, содержащие только 4 буквы: X, Y, Z, W; для кодировки букв используются кодовые слова длины 5. При этом для набора кодовых слов выполнено такое свойство: *любые два слова из набора отличаются не менее чем в трёх позициях*. Это свойство важно для расшифровки сообщений при наличии помех. Для кодирования букв X, Y, Z используются 5-битовые кодовые слова:
X: 01111, Y: 00001, Z: 11000. Определите 5-битовое кодовое слово для буквы W, если известно, что оно начинается с 1 и заканчивается 0. *Ответ запишите в строке под заданием.*

---

6 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 минимальное число N, после обработки которого с помощью этого алгоритма получается число, большее, чем 137. В ответе это число запишите в десятичной системе. *Ответ запишите в строке под заданием.*

---

7 Дан фрагмент электронной таблицы. Из ячейки D2 в одну из ячеек диапазона E1:E4 была скопирована формула. При копировании адреса ячеек в формуле автоматически изменились, и значение формулы стало равным 8. В

какую ячейку была скопирована формула? В ответе укажите только одно число – номер строки, в которой расположена ячейка. *Ответ запишите в строке под заданием.*

	A	B	C	D	E
1	1	2	3	4	
2	2	3	4	=B$3+$C2	
3	3	4	5	6	
4	4	5	6	7	

---

8 Определите, что будет напечатано в результате работы следующего фрагмента программы. *Ответ запишите в строке под заданием.*

Бейсик

```
DIM S, K AS INTEGER
S = 0
K = 1
WHILE S < 11
S = S + K
K = K + 1
WEND
PRINT S
```

Python

```
s=0
k=1
while k < 11:
  s=s+k
  k=k+1
print(s)
```

Алгоритмический язык

```
алг
нач
цел k, s
s := 0
k := 1
нц пока k < 11
s := s + k
k := k + 1
кц
вывод s
ко
```

Паскаль

```
var k, s: integer;
begin
  s:=0;
  k:=1;
  while k < 11 do begin
    s:=s+k;
    k:=k+1;
  end;
  write(s);
end.
```

C++

```
#include <iostream>
using namespace std;
int main() {
  int k, s;
  s = 0;
  k = 1;
  while (k < 11) {
    s = s + k;
    k = k + 1;
  }
  cout << s;
  return 0;
```

```
}
```

---

9 Рисунок размером 512 на 256 пикселей занимает в памяти 64 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения. *Ответ запишите в строке под заданием.*

---

10 Все 4-буквенные слова, составленные из букв К, Л, Р, Т, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. КККК
2. КККЛ
3. КККР
4. КККТ

……

Запишите слово, которое стоит на 67-м месте от начала списка. *Ответ запишите в строке под заданием.*

---

11 Ниже на пяти языках программирования записан рекурсивный алгоритм F.

Бейсик

```
SUB F(n)
    IF n > 0 THEN
        PRINT n
        F(n - 3)
        F(n \ 3)
    END IF
END SUB
```

Python

```
def F(n):
    if n > 0:
        print(n)
        F(n - 3)
        F(n // 3)
```

Алгоритмический язык

```
алг F(цел n)
нач
    если n > 0 то
        вывод n
        F(n - 3)
        F(div(n, 3))
    все
кон
```

Паскаль

```
procedure F(n: integer);
begin
    if n > 0 then
    begin
        write(n);
        F(n - 3);
        F(n div 3)
    end
end;
```

C++

```
void F(int n){
    if (n > 0){
        std::cout <<n;
        F(n - 3);
        F(n / 3);
```

```
    }
}
```

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.
Ответ запишите в строке под заданием.

---

12 Два узла, находящиеся в одной сети, имеют IP-адреса 118.222.130.140 и 118.222.201.140. Укажите наибольшее возможное значение третьего слева байта маски сети. Ответ запишите в виде десятичного числа. *Ответ запишите в строке под заданием.*

---

13 При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 11 символов. Из соображений информационной безопасности каждый пароль должен содержать хотя бы 2 десятичных цифры, как прописные, так и строчные латинские буквы, а также не менее 2-х символов из 6-символьного набора: «&», «#», «$», «*», «!», «@». В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей. Для хранения сведений о 30 пользователях потребовалось 900 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число – количество байт.

---

14 Редактор получает на вход строку цифр и преобразовывает её. Редактор может выпол-нять две команды, в обеих командах v и w обозначают цепочки цифр.

А) заменить (v, w)

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w.

Б) нашлось (v)

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка при этом не изменяется.
Дана программа для исполнителя Редактор:

```
НАЧАЛО
ПОКА нашлось (222) ИЛИ нашлось (888)
  ЕСЛИ нашлось (222)
    ТО заменить (222, 8)
```

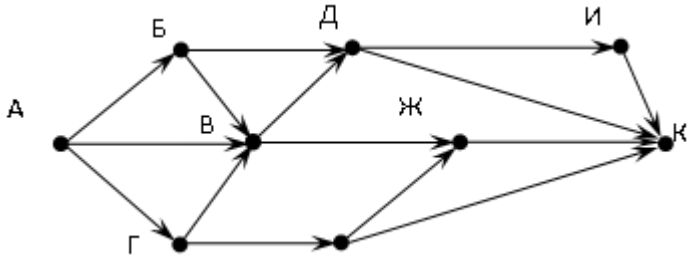
ИНАЧЕ заменить (888, 2)
КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ

Какая строка получится в результате применения приведённой выше программы к строке, состоящей из 68 идущих подряд цифр 8? В ответе запишите полученную строку. *Ответ запишите в строке под заданием.*

---

15 На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?

Ответ запишите в строке под заданием.



---

16 Значение арифметического выражения: $9^9 - 3^9 + 9^{19} - 19$ записали в системе счисления с основанием 3. Сколько цифр «2» содержится в этой записи?

Ответ запишите в строке под заданием.

---

17 В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет:

Запрос	***Количество страниц (тыс.)***
Ростов & (Орёл & Курск | Белгород)	*370*
Ростов & Белгород	*204*
Ростов & Орёл & Курск & Белгород	*68*

Сколько страниц (в тысячах) будет найдено по запросу

Ростов & Орёл & Курск

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов. *Ответ запишите в строке под заданием.*

---

18 Укажите наименьшее целое значение А, при котором выражение

$$(y + 2x < A) \vee (3y + 2x > 120) \vee (3y - x > 30)$$

истинно для любых целых положительных значений x и y. *Ответ запишите в строке под заданием.*

19

В программе используется одномерный целочисленный массив A с индексами от 0 до 9. Значения элементов равны 3, 0, 4, 6, 5, 1, 8, 2, 9, 7 соответственно, т.е. A[0] = 3, A[1] = 0 и т.д.

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

Бейсик

```
c = 0
FOR i = 1 TO 9
    IF A(i-1) > A(i) THEN
        c = c + 1
        t = A(i)
        A(i) = A(i-1)
        A(i-1) = t
    END IF
NEXT i
```

Python

```
c = 0
for i in range(1,10):
    if A[i-1] > A[i]:
        c = c + 1
        t = A[i]
        A[i] = A[i-1]
        A[i-1] = t
```

Алгоритмический язык

```
c := 0
нц для i от 1 до 9
    если A[i-1] > A[i] то
        c := c + 1
        t := A[i]
        A[i] := A[i-1]
        A[i-1] := t
    все
кц
```

Паскаль

```
c := 0;
for i := 1 to 9 do
    if A[i-1] > A[i] then
    begin
        c := c + 1;
        t := A[i];
        A[i] := A[i-1];
        A[i-1] := t;
    end;
```

C++

```
c = 0;
for (int i = 1; i < 10; i++)
    if (A[i-1] > A[i]){
        c++;
        t = A[i];
        A[i] = A[i-1];
        A[i-1] = t;
    }
```

Ответ запишите в строке под заданием.

_____

20. Ниже на пяти языках программирования записан алгоритм. Получив на вход число x, этот алгоритм печатает два числа: L и M. Укажите наименьшее число x, при вводе которого алгоритм печатает сначала 5, а потом 7.

Бейсик

```
DIM X, L, M AS INTEGER
INPUT X
```

Python

```
x = int(input())
L = 0
```

```
L = 0
M = 0
WHILE X > 0
        M = M + 1
        IF X MOD 2 <> 0 THEN
                L = L + 1
        END IF
X = X \ 2
WEND
PRINT L
PRINT M
```

```
M = 0
while x > 0:
        M = M + 1
        if x % 2 != 0:
                L = L + 1
        x = x // 2
print(L)
print(M)
```

Алгоритмический язык

```
алг
нач
        цел x, L, M
        ввод x
        L := 0
        M := 0
        нц пока x > 0
                M := M + 1
                если mod(x,2) <> 0
                        то
                        L := L + 1
                все
                x := div(x,2)
        кц
        вывод L, нс, M
кон
```

Паскаль

```
var x, L, M: integer;
begin
        readln(x);
        L := 0;
        M := 0;
        while x>0 do
        begin
                M := M + 1;
                if x mod 2 <> 0 then
                        L := L + 1;
                x := x div 2;
        end;
        writeln(L);
        writeln(M);
end.
```

C++

```
#include <iostream>
using namespace std;
int main(){
	int x, L, M;
	cin >> x;
	L = 0;
	M = 0;
	while (x > 0) {
		M = M + 1;
		if(x % 2 != 0) {
			L = L + 1;
		}
		x = x / 2;
	}
	cout << L << endl << M << endl;
	return 0;
}
```

Ответ запишите в строке под заданием.

_____

21 Напишите в ответе число, которое будет напечатано в результате выполнения следующего алгоритма. Для Вашего удобства алгоритм представлен на пяти языках программирования.

Бейсик

```
DIM A, B, T, M, R AS LONG
A = -20: B = 20
M = A: R = F(A)
FOR T = A TO B
	IF F(T) <= R THEN
		M = T
		R = F(T)
	END IF
NEXT T
PRINT M+R
FUNCTION F (x)
	F = 2*(x*x-1)*(x*x-1)+27
END FUNCTION
```

Python

```
def F(x):
	return 2*(x*x-1)*(x*x-1)+27
a = -20; b=20
M=a; R=F(a)
for t in range(a,b+1):
	if (F(t) <= R):
		M=t; R=F(t)
print (M+R)
```

Алгоритмический язык	**Паскаль**

алг
нач
цел a, b, t, M, R
a:=-20; b:=20
M:=a; R:=F(a)
нц для t от a до b
если F(t) <= R то
M:=t; R:=F(t)
все
кц
вывод M+R
кон
алг цел F(цел x)
нач
знач:=2*(x*x-1)*(x*x-1)+27
кон
	var a, b, t, M, R :longint;
	function F(x: longint) :
	longint;
	begin
	F:= 2*(x*x-1)*(x*x-1)+27;
	end;
	begin
	a:=-20; b:=20;
	M:=a; R:=F(a);
	for t:= a to b do begin
	if (F(t) <= R) then begin
	M:=t;
	R:=F(t)
	end
	end;
	write(M+R)
	end.

C++

```
#include <iostream>
using namespace std;
long F(long x) {
	return 2*(x*x-1)*(x*x-1)+27;
}
int main() {
	long a = -20, b = 20, M = a, R = F(a);
	for (int t = a; t <= b; ++t) {
		if (F(t)<= R) {
			M = t; R = F(t);
		}
	}
	cout << M + R;
	return 0;
}
```

Ответ запишите в строке под заданием.

---

22 Исполнитель M17 преобразует число, записанное на экране.

У исполнителя есть три команды, которым присвоены номера:

1. Прибавить 1

2. Прибавить 2

3. Умножить на 3

Первая из них увеличивает число на экране на 1, вторая увеличивает его на 2, третья умножает на 3.

Программа для исполнителя M17 – это последовательность команд.

Сколько существует таких программ, которые преобразуют исходное число 2 в число 12 и при этом траектория вычислений программы содержит числа 8 и 10? Траектория должна содержать оба указанных числа.
Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы **132** при исходном числе 7 траектория будет состоять из чисел 8, 24, 26.
Ответ запишите в строке под заданием.

---

23 Сколько различных решений имеет система логических уравнений

$$(x_1 \wedge y_1) \rightarrow (\neg x_2 \vee \neg y_2) = 1$$
$$(x_2 \wedge y_2) \rightarrow (\neg x_3 \vee \neg y_3) = 1$$
$$(x_3 \wedge y_3) \rightarrow (\neg x_4 \vee \neg y_4) = 1$$
$$(x_4 \wedge y_4) \rightarrow (\neg x_5 \vee \neg y_5) = 1$$
$$x_2 \wedge y_4 = 0$$

где $x_1, \ldots, x_5, y_1, \ldots, y_5$ – логические переменные? В ответе не нужно перечислять все различные наборы значений переменных, при которых выполнено данное равенство. В качестве ответа нужно указать количество таких наборов.
Ответ запишите в строке под заданием.

Часть 2

24 Требовалось написать программу, в которой нужно было проверить, лежит ли число *x* на числовой оси между числами *a* и *b* ("между" понимается в строгом смысле, т.е. случай *x=a* или *x=b* недопустим). Числа *x, a, b* являются натуральными, и известно, что *a* отлично от *b* (но неизвестно: *a>b* или *b>a*). Входная информация вводится с клавиатуры, а на выходе должно быть сообщение вида "*x* между *a* и *b*" (если это действительно так), в противном случае никакой выходной информации не выдается.

Программист торопился и написал программу некорректно.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ

VAR a,b,x: integer;
p: integer;
BEGIN
readln(a,b,x);
if (a>x) AND (x>b) then
writeln('x между a,b');
END.
	CLS
	INPUT a, b, x
	IF (a>x) AND (x>b) THEN
	PRINT “x между a, b”
	END

Последовательно выполните три задания. *Ответы впишите в поле под заданиями.*

1) Приведите пример таких чисел *a, b, x*, при которых программа работает неправильно.

2) Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы).

3) Укажите, как можно доработать программу, соблюдая дополнительное условие: доработанная программа не должна использовать логических операций AND или OR.

1.

2.

3.

25 Опишите на русском языке или на одном из языков программирования алгоритм поиска второго по величине (т.е. следующего по величине за максимальным) элемента в числовом массиве из 30 различных элементов.
Ответ впишите в поле под заданием.

26 Два игрока играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 5, а во второй – 3 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди. Ход состоит в том, что игрок или удваивает число камней в какой-то куче, или добавляет 4 камня в какую-то кучу. Выигрывает игрок, после хода которого в одной из куч становится не менее 22 камней. Кто выигрывает при безошибочной игре обоих игроков – игрок, делающий первый ход, или игрок, делающий второй ход? Как должен ходить выигрывающий игрок? Ответ обоснуйте. *Ответ впишите в поле под заданием.*

27 Вступительные испытания в некоторый вуз состоят из трех экзаменов: математика (максимальный балл – 9), информатика (максимальный балл – 9), литература (максимальный балл – 5). На вход программе подаются сведения о сдаче этих экзаменов абитуриентами. В первой строке вводится количество абитуриентов N, во второй – количество мест K (K < N) на которые эти абитуриенты претендуют. Каждая из следующих N строк имеет следующий формат: <Фамилия> <оценка1> <оценка2> <оценка3>, где <Фамилия> – строка, состоящая не более, чем из 20 символов, оценки – числа от 0 до максимальной оценки по предмету соответственно. (Ноль ставится в случае, если экзамен не

сдавался, например, после полученной на предыдущем экзамене двойки. Все баллы, большие 2, считаются удовлетворительными). Пример входных строк:
Иванов 8 9 3
Петров 2 0 0
Требуется написать программу на языке Паскаль или Бейсик, которая определяла бы по имеющимся данным количество абитуриентов, набравших полупроходной балл в данный вуз или сообщала, что такой балл отсутствует. (Полупроходным называется такой балл, что лишь часть абитуриентов, набравших такой балл и не получивших ни одной неудовлетворительной оценки, попадает в К лучших, которые должны быть зачислены на 1 курс) Считается, что абитуриенты, получившие только удовлетворительные оценки, обязательно присутствуют. *Ответ впишите в поле под заданием.*

ИНСТРУКЦИЯ ПО ПРОВЕРКЕ И ОЦЕНКЕ РАБОТ

ЧАСТЬ 1

За правильный ответ на задания 1–23 ставится 3 балла; за неверный ответ или его отсутствие – 0 баллов.

№ задания	Ответ
1	**7**
2	**zxy**
3	**20**
4	**7**
5	**10110**
6	**35**
7	**1**
8	**55**
9	**16**
10	**ЛККР**
11	**9631231**
12	**128**
13	**20**
14	**28**
15	**13**
16	**16**
17	**234**
18	**1188**
19	**5**
20	**79**
21	**28**
22	**60**
23	**256**

Итого за Часть 1: 69 баллов

ЧАСТЬ 2

24

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	**Баллы**
Элементы ответа: 1) Пример: a=1 x=2 b=3 2) Возможная доработка: if a<b then begin p:=a; a:=b; b:=p end; if (a>x) AND (x>b) then writeln(' x между a,b'); (могут быть и другие правильные способы доработки). 3) Возможная доработка без использования логических операций AND, OR: p:=(x-a)*(x-b); if p<0 then writeln(' x между a,b'); (могут быть и другие способы доработки с соблюдением дополнительного условия). При оценке других вариантов доработки программы нужно проверять, что поставленная цель достигается.	
Указания по оцениванию	
Правильно выполнены п.1)+п.3) задания (т.к. выполнение п.3 "покрывает" и пункт 2), или правильно выполнены все 3 пункта задания, при этом в работе (во фрагментах программ) допускается не более одной пунктуационной ошибки	6
Правильно выполнены 2 пункта задания:1)+2) или 2)+3), (причем способы доработки в п.2 и п.3 различны). При этом в сданной работе допускается не более двух синтаксических ошибок (пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования)	4
Правильно выполнен только один пункт задания, при этом, если это был п.2) или п.3), то в нем допускается не более двух синтаксических ошибок (пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования)	2
Все пункты задания выполнены неверно	0
Максимальный балл	6

25

<table>
<tr><th>Содержание верного ответа и указания по оцениванию
(допускаются иные формулировки ответа, не искажающие его смысла)</th><th>Баллы</th></tr>
<tr><td>Введем числовые переменные Max1 и Max2, в которых будем хранить соответственно максимальный и следующий за максимальным элемент в уже просмотренной части массива. Затем в цикле до конца массива сравниваем очередной элемент массива с двумя максимальными, и если он больше одного из них или обоих, то меняем два отобранных элемента. По окончании цикла переменная Max2 содержит второй по величине элемент массива.</td><td></td></tr>
<tr><td>Указания по оцениванию</td><td></td></tr>
<tr><td>Предложен правильный алгоритм, выдающий верное значение (в том числе и алгоритм, требующий двукратного прохода по массиву).
Возможно использование числа 30 вместо константы.
Возможно наличие отдельных синтаксических ошибок (пропущенные «;», неверная запись оператора присваивания и т.п.), не искажающих замысла автора программы.
В качестве примера правильного и эффективного алгоритма приведен фрагмент программы:
<table>
<tr><td>На языке Паскаль</td><td>На языке Бейсик</td></tr>
<tr><td><pre>const N=30;
var a:array[1..N] of real;
   Max1, Max2, i: real;
begin
 Max1:=a[1];
 Max2:=a[1];
 if a[2]>Max1 then Max1:=a[2]
          else Max2:=a[2];
 for i:=3 to N do
  begin
   if a[i]>Max1 then
     begin Max2:=Max1;
          Max1:=a[i];
     end
     else if a[i]>Max2 then
          Max2:=a[i];
  end;
  writeln(Max2);
end.</pre></td><td><pre>N=30
DIM i, Max1, Max2, a(N) AS REAL
Max1=a(1)
Max2=a(1)
IF a(2)>Max1 THEN Max1=a(2)
ELSE Max2=a(2)
FOR i = 3 TO N
IF a(i)>Max1 THEN
            Max2=Max1
            Max1=a(i)
          ELSE
          IF a(i)>Max2 THEN
             Max2=a(i)
          ENDIF
  ENDIF
NEXT i
PRINT Max2
END</pre></td></tr>
</table></td><td>6</td></tr>
<tr><td>Имеется не более двух ошибок из числа следующих:
1) Не задано или неверно задано первое значение Max1
2) Неверно вычисляется первое значение переменной Max2
3) Не указано условие завершения цикла
4) Программа не выводит результат</td><td>3</td></tr>
</table>

5) Индексная переменная в цикле не увеличивается (при использовании циклов while или repeat-until) В программе на Паскале неверно расставлены операторные скобки	
Ошибок, перечисленных выше, больше двух или алгоритм сформулирован неверно (в частности, не хранится следующий за максимальным элемент).	0
Максимальный балл	6

26

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	**Баллы**

Выигрывает первый игрок. Своим первым ходом он должен удвоить количество камней во второй куче. Для доказательства рассмотрим неполное дерево игры после этого хода первого игрока.

Позиция после первого хода	1-й ход второго игрока	Выигрывающий ход первого игрока	Пояснение
	10,6	10,10	Первый игрок выигрывает после любого ответа второго игрока, удвоив число камней в самой большой куче
5,6	9,6	9,10	-"-
	5,10	9,10 или 10,10	-"-
	5,12	5,24	Выигрыш первого игрока

Из таблицы видно, что при первом ходе (5,3)->(5,6) первый игрок выигрывает не позже, чем на третьем ходу при любом ответе второго игрока.

Указания по оцениванию	
Правильное указание игрока и его ходов со строгим доказательством правильности с помощью или без помощи дерева игры	9
Правильное указание выигрывающего игрока, стратегии игры, приводящей к победе, но при отсутствии доказательства ее правильности.	6

При наличии в представленном решении одного из пунктов: 1. Правильно указаны все варианты хода первого игрока и возможные ответы второго игрока (в том числе, и все выигрышные), но неверно определены дальнейшие действия и неправильно указан победитель. 2. Правильно указан выигрывающий игрок, но отсутствует описание выигрышной стратегии и рассмотрены не все варианты хода первого игрока (а только один или несколько) и частные случаи ответов второго игрока.	3
Задание не выполнено или в представленном решении полностью отсутствует описание элементов выигрышной стратегии, и отсутствует анализ вариантов первого и второго ходов играющих.	0
Максимальный балл	9

27

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	**Баллы**
Программа верно читает входные данные, не запоминая их все, а сразу подсчитывая в массиве, хранящем 24 целых числа, количество абитуриентов, набравших тот или иной балл (от 0 до 23). Если при этом абитуриент получил хотя бы одну двойку, то удобно считать, что его общий балл равен 0. Затем вычисляется сумма элементов этого массива, начиная с 23-го, до тех пор пока она не превосходит K. Индекс первого элемента массива, который не войдет в эту сумму и будет искомым полупроходным баллом. Если проходной балл набрали ровно K абитуриентов, то программа сообщает, что полупроходной балл отсутствует. Баллы начисляются только за программу, которая решает задачу хотя бы для частного случая (например, проходной балл набрали строго меньше K абитуриентов).	
Указания по оцениванию	
Программа работает верно, т.е. корректно выделяет из входных данных оценки абитуриентов, верно учитывает результаты абитуриентов, получивших двойки, не содержит вложенных циклов (от 1 до N и от 0 до 23). Допускается наличие в тексте программы одной пунктуационной ошибки.	10

Пример правильной и эффективной программы на языке Паскаль:

```
var m:array[0..23] of integer;
    c:char;
    i, K, N, S, m1, m2, m3:integer;
begin
  readln(N); readln(K);
  for i:=0 to 23 do m[i]:=0;
  for i:=1 to N do
  begin
    repeat
      read(c)
    until c=' '; {считана фамилия абитуриента}
    readln(m1, m2, m3);
    if (m1<3)or(m2<3)or(m3<3) then s:=0
    else s:=m1+m2+m3;
    m[s]:=m[s]+1  {учитываем абитуриента в элементе
массива, соответствующем его баллам}
  end;
  s:=m[23]; i:=23;
  while s+m[i-1]<=K and
        (i>9) {9 - минимально возможный балл} do
  begin
    i:=i-1;
    s:=s+m[i]
  end;
  if (s<K)and(i>9) then
       writeln('полупроходной балл набрали', m[i-1],
               ' человек')
  else writeln('полупроходной балл отсутствует');
  readln
end.
```

Пример правильной программы на языке Бейсик:
`DIM i, j, k, n, m1, m2, m3, s, m(23) AS INTEGER`
`DIM ss AS STRING`
`FOR i = 1 TO 23`
`  m(i) = 0`
`NEXT i`
`INPUT n`
`INPUT k`
`FOR j = 1 TO n`
`LINE INPUT ss`
`i = 1`
`c$ = MID$(ss, i, 1)`
`WHILE NOT (c$ = " ")`
`  i = i + 1`
`  c$ = MID$(ss, i, 1)`
`WEND`
`ss = MID$(ss, i + 1, 5)`
`m1 = ASC(MID$(ss, 1, 1)) - ASC("0")`
`m2 = ASC(MID$(ss, 3, 1)) - ASC("0")`
`m3 = ASC(MID$(ss, 5, 1)) - ASC("0")`
`IF (m1 < 3) OR (m2 < 3) OR (m3 < 3) THEN`
`  s = 1`
`  ELSE s = m1 + m2 + m3`
`END IF`
`m(s) = m(s) + 1`
`NEXT j`
`s = m(23): i = 23`
`WHILE (s + m(i - 1) <= k) AND (i > 9)`
`  i = i - 1`
`  s = s + m(i)`
`WEND`
`IF (s < k) AND (i > 9) THEN`
`PRINT "Полупроходной балл набрали"; m(i - 1);" человек"`
`ELSE PRINT "Полупроходной балл отсутствует"`
`END IF`
`END`	

Программа работает верно, но содержит вложенные циклы (от 0 до 23 и от 1 до N) или несколько операторов IF (по количеству возможных баллов у абитуриента) или оператор CASE, обрабатывающий различные варианты количества баллов абитуриента. Возможно, сохраняет все входные данные в массиве абитуриентов. Допускается наличие от одной до трех различных синтаксических ошибок: пропущен или неверно указан знак пунктуации, неверно написано зарезервированное слово языка программирования, не описана или неверно описана переменная, применяется операция, недопустимая для соответствующего типа данных.	8

Программа не учитывает случай, когда ровно K абитуриентов набрали проходной балл или что количество абитуриентов, получивших удовлетворительные оценки, может оказаться меньше K. Возможно, в реализации алгоритма содержатся 1–2 ошибки (используется знак “<” вместо “<=”, “or” вместо “and”, выражение на 1 отличается от верного и т.п.). Допускается наличие от одной до пяти различных синтаксических ошибок.	6
Программа неверно работает при некоторых входных данных, возможно, содержит ошибку при учете баллов абитуриентов, получивших неудовлетворительные оценки или в выделении оценок из строки входных данных или в логике определения полупроходного балла. Допускается наличие от одной до семи различных синтаксических ошибок.	3
Задание выполнено неверно	0
Максимальный балл	10

Итого за Часть 2: 31 баллов