

Государственное образовательное учреждение высшего образования
«КОМИ РЕСПУБЛИКАНСКАЯ АКАДЕМИЯ ГОСУДАРСТВЕННОЙ СЛУЖБЫ И
УПРАВЛЕНИЯ»
(ГОУ ВО КРАГСнУ)

«КЛИМУ СЛУЖБАӦ ДА ВЕСЬКӦДЛЫНЫ ВЕЛӦДАН КОМИ РЕСПУБЛИКАСА
АКАДЕМИЯ»

вылыс тшупӧда велӧдан канму учреждение
(КСдаВВКРА ВТШВ КУ)

УТВЕРЖДЕНО
Решением Приемной комиссии
от 29.10.2021 № 1



**ПРОГРАММА ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ
ПО ИНФОРМАТИКЕ И
ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫМ ТЕХНОЛОГИЯМ**

Сыктывкар
2021

I. Структура вступительного испытания

Время выполнения заданий – 235 минут.

Вступительное испытание по информатике и информационно-коммуникационным технологиям состоит из двух частей, включающих 27 заданий.

Форма заданий:

- часть I содержит 23 задания с кратким ответом нескольких разновидностей:
 - задания на выбор и запись одного или нескольких правильных ответов из предложенного перечня ответов;
 - задания на вычисление определенной величины;
 - задания на установление правильной последовательности, представленной в виде строки символов по определенному алгоритму.
- часть II содержит 4 задания. Задания этой части подразумевают запись развернутого ответа в произвольной форме.

Шкала оценивания:

№ задания	балл
1-24	1
25-27	2

Шкала перевода первичного балла в стобальную систему:

Первичный	1	2	3	4	5	6	7	8	9	10
Тестовый	8	15	22	29	37	43	46	48	50	53
Первичный	11	12	13	14	15	16	17	18	19	20
Тестовый	55	57	60	62	65	67	69	71	74	76
Первичный	21	22	23	24	25	26	27	28	29	30
Тестовый	78	81	84	85	88	90	93	95	98	100

II. Содержание программы

1. Информация и её кодирование. Виды информационных процессов. Свойства информации. Количество информации. Единицы измерения информации. Формула Хартли. Процесс передачи информации, источник и приемник информации. Сигнал, кодирование и декодирование. Дискретное (цифровое) представление текстовой, графической, звуковой информации и видеоинформации. Скорость передачи информации и пропускная способность канала передачи. Представление и кодирование информации с помощью знаковых систем. Хранение информации в компьютере. Кодирование текстовой информации (ASCII, 8-битные кодировки, таблица символов Unicode и связанные с ней кодировки). Кодирование графической информации. Цветовая модель RGB и другие варианты цветового кодирования. Кодирование звуковой информации, глубина кодирования и частота дискретизации.

2. Системы счисления и основы логики. Знакомство с различными системами счисления. Позиционные и непозиционные системы счисления. Перевод целых чисел из 10-чной системы счисления в 2-чную, 8-чную, 16-чную и обратно. Арифметические операции в двоичной системе счисления. Арифметические операции в различных позиционных системах счисления. Основы логики. Основные логические элементы (отрицание, дизъюнкция, конъюнкция, импликация, эквиваленция), их таблицы истинности. Законы логики. Восстановление логических функций по таблице истинности, упрощение логических выражений, решение логических задач на определение истинности набора высказываний.

3. Алгоритмы и элементы программирования. Понятие алгоритма. Свойства алгоритма. Формы записи алгоритма: естественно-языковая, графическая (на языке блок-схем), на языках программирования. Основные структуры алгоритмов. Примеры линейных, ветвящихся и циклических алгоритмов. Этапы подготовки и решения задач на компьютере.

Базовые понятия языка программирования: константы и переменные, типы данных и арифметические операции, ввод-вывод, оператор присваивания. Управляющие операторы и их виды (условный оператор, оператор выбора, цикл с параметром, цикл с предусловием, цикл с постусловием). Вложенные циклы. Представление управляющих операторов в виде блок-схем. Процедуры и функции: описание и использование. Параметры, виды их передачи. Массивы, их свойства. Одномерные и двумерные массивы. Базовые алгоритмы обработки массивов: создание массива, поиск элементов массива по заданным признакам, накопление суммы, произведения, подсчет количества элементов массива, удовлетворяющих заданным условиям, сортировка массива. Обработка символьной информации. Работа с записями. Система программирования. Трансляторы языков: интерпретаторы и компиляторы. Основные этапы разработки программ. Разбиение задачи на подзадачи.

4. Обработка числовой информации. Математическая обработка статистических данных. Электронные таблицы: назначение и основные возможности. Редактирование структуры таблицы. Абсолютная и относительная адресация ячеек. Ввод чисел, формул и текста. Стандартные функции. Основные объекты в электронных таблицах и операции над ними (ячейка, лист, книга). Построение диаграмм. Использование электронных таблиц для решения задач. Визуализация данных с помощью диаграмм и графиков. Использование инструментов решения статистических и расчетно-графических задач.

5. Технологии создания и обработки текстовой информации. Создание компьютерных публикаций. Использование готовых и создание собственных шаблонов. Использование систем проверки орфографии и грамматики. Тезаурусы. Использование систем двуязычного перевода и электронных словарей. Использование специализированных средств редактирования математических текстов и графического представления математических объектов. Использование систем распознавания текстов. Правила набора текста, копирование, перемещение и удаление фрагментов текста. Форматирование абзацев. Создание и редактирование таблиц, изменение направления текста, выравнивание информации в ячейках таблицы.

6. Технология обработки графической информации. Представление графической информации. Пиксель. Графические примитивы. Способы хранения графической информации и форматы графических файлов. Графический редактор: назначение, пользовательский интерфейс и основные возможности. Графические объекты и операции над ними.

7. Поиск и хранение информации. Базы данных: основные понятия. Типы баз данных. Реляционные базы данных. Системы управления базами данных. Ввод и редактирование записей. Сортировка и поиск записей. Основные объекты в базах данных и операции над ними (запись, поле). Изменение структуры базы данных. Виды и способы организации запросов.

8. Коммуникационные технологии. Программная и аппаратная организация компьютерных систем. Виды программного обеспечения. Локальные и глобальные компьютерные информационные сети. Основные информационные ресурсы: электронная

почта, телеконференции, файловые архивы. Гипертекст. Интернет. Технология World Wide Web (WWW). Публикации в WWW. Поиск информации. IP-адреса и доменные имена. Адресация в сети. Маска подсети. Поисковые системы в компьютерных сетях. Правила построения запросов. Личное информационное пространство пользователя в сети Интернет. Сетевые сообщества. Публикация материалов в сети Интернет. Использование информационных систем в Интернете. Электронная коммерция. Разработка веб-сайтов. Язык HTML. Специальное программное обеспечение средств телекоммуникационных технологий. Инструменты создания информационных объектов для Интернета. Понятие о системном администрировании.

9. Компьютер и программное обеспечение. Основные устройства персонального компьютера, их назначение и краткая характеристика: процессор, оперативная память, внешняя энергонезависимая память, устройства ввода-вывода. Программное обеспечение компьютера. Системное и прикладное программное обеспечение. Операционная система: назначение и основные функции. Файлы и каталоги. Работа с носителями информации. Представление об объёмах данных и скоростях доступа, характерных для различных видов носителей. История и тенденции развития компьютеров, улучшение характеристик компьютеров, физические ограничения на значения характеристик. Суперкомпьютеры.

10. Моделирование и формализация. Моделирование как метод познания. Материальные и информационные модели. Формализация. Математические модели. Логические модели. Детерминированные и вероятностные модели. Расчётные модели. Оптимизационные модели. Имитационные модели. Построение и исследование на компьютере информационных моделей из различных предметных областей.

III. Рекомендуемый библиографический список

1. Информатика. Углубленный уровень: учебник для 10 класса: в 2 ч. Ч. 1, Ч. 2 / И. Г. Семакин, Т. Ю. Шеина, Л. В. Шестакова. – Москва: БИНОМ. Лаборатория знаний, 2014. – 184 с.
2. Информатика и ИКТ. Базовый уровень: учебник для 10 класса / И. Г. Семакин, Е. К. Хеннер, Т. Ю. Шеина. – 4-е. изд. – Москва: БИНОМ, Лаборатория знаний, 2015. – 264 с.
3. ЕГЭ 2017. Информатика. Тематические тестовые задания / С. С. Крылов, Д. М. Ушаков. – Москва: Издательство «Экзамен», 2017. – 271 с.
4. Турбо Паскаль 7.0 – Киев: Издательская группа BHV, 1998 – 448 с.

IV. Примерный тест по информатике и ИКТ

1. Сколько существует натуральных чисел x , для которых выполнено неравенство $346_8 < x < EA_{16}$? В ответе укажите только количество чисел, сами числа писать не нужно.
2. Миша заполнял таблицу истинности функции $(\neg x \wedge \neg y) \vee (y \equiv z) \vee \omega$, но успел заполнить лишь фрагмент из трех различных ее строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных ω, x, y, z .

Перем. 1	Перем. 2	Перем. 3	Перем. 4	Функция
???	???	???	???	F
1			1	0
1	0	0		0

		1	1	0
--	--	---	---	---

В отчете запишите буквы x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала – буква, соответствующая 1-му столбцу; затем – буква, соответствующая 2-му столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример: Если бы функция была задана выражением $x \rightarrow y$, зависящее от двух переменных x и y, а фрагмент таблицы имел бы вид:

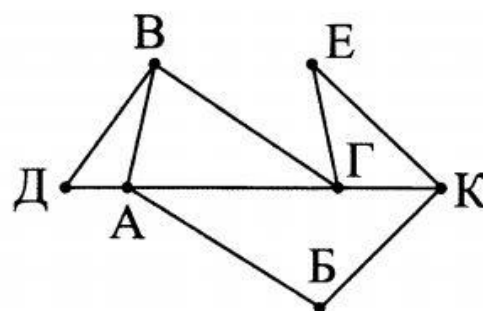
		$x \rightarrow y$
0	1	0

то первому столбцу соответствовала бы переменная y, а второму столбцу – переменная x.

В ответе следовало бы написать: ух.

3. На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

	П1	П2	П3	П4	П5	П6	П7
П1		35	15		55	20	
П2	35			30	45		50
П3	15					10	25
П4		30			40		
П5	55	45		40			
П6	20		10				
П7		50	25				



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населенных пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта Б в пункт К. В ответе запишите целое число – так, как оно указано в таблице.

4. Дан фрагмент базы данных, содержащий информацию о родственных отношениях. Определите на основании приведенных данных ID человека, который стал отцом в самом молодом возрасте.

Таблица 1

ID	Фамилия И.О.	Пол	Год рождения
145	Дени К.Д.	Ж	1957
218	Ирта О.Б.	Ж	1978
260	Гано И.А.	Ж	1980
347	Ивенко Д.Я.	М	1936
421	Бабенко А.Е.	М	1994
453	Старых Г.В.	Ж	1919
536	Марченко Е.А.	М	1975
577	Арне А.А.	М	1921
619	Рудко К.А.	Ж	2019
689	Шевченко А.В.	М	2002
714	Сиро А.К.	М	1955
781	Исава Г.А.	Ж	1940
820	Сорт А.К.	Ж	2000
907	Смирных Н.Д.	Ж	1960

Таблица 2

ID_Родителя	ID_Ребёнка
536	421
218	421
714	536
145	536
347	145
453	781
781	145
714	260
145	260
347	907
781	907
260	820
577	781
689	619
260	689

5. Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, используется неравномерный код, удовлетворяющий Фано. Вот этот код: А-01, Б-

- 11, В-00, Г-101. Каким кодовым словом должна кодироваться буква Д? Если таких вариантов несколько, укажите самый короткий.
6. На вход алгоритма подается натуральное число N. Алгоритм строит по нему новое число R следующим образом.
- 1) Строится двоичная запись числа N.
 - 2) К этой записи дописываются разряды по следующему правилу:
 - а) если число четное, то к двоичной записи числа в начало дописывается 1, а в конец дописывается 0;
 - б) если число нечетное, то к двоичной записи числа в начало дописывается 11, и в конец дописывается 11.

Полученная таким образом запись является двоичной записью искомого числа R.

Укажите минимальное число R, которое превышает 100 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

7. Дан фрагмент электронной таблицы
- Чему станет равным значение ячейки D2, если в нее скопировать формулу из ячейки C3?

	A	B	C	D
1	1	2	3	
2	5	6	=A2+B1	
3	7	8	=B\$1+B3	

Примечание: знак \$ обозначает абсолютную адресацию.

8. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 1 s = 8 WHILE k < 500 s = s + 20 k = k * 5 WEND PRINT s </pre>	<pre> var k, s : integer; begin k := 1; s := 8; while k < 500 do begin s := s + 20; k := k * 5 end; write(s) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> int main() { int k = 1, s = 8; while (k < 500) { s = s + 20; k = k * 5; } std::cout << s; return 0 } </pre>	<pre> нач цел k, s k := 1 s := 8 нц пока k < 500 s := s + 20 k := k * 5 кц вывод s кон </pre>
Python	
<pre> k = 1; s = 8 while k < 500: s = s + 20 k = k * 5 print(s) </pre>	

9. Автоматическая фотокамера производит растровые изображения размером 2400 на 1600 пикселей. При этом объем файла с изображением не может превышать 2500 Кбайт, упаковка данных не производится. Какое максимальное количество цветов можно использовать в палитре изображения?

10. Для кодирования 500 различных сообщений используют 4 последовательных цветовых вспышки. Лампочки скольких различных цветов должны использоваться при передаче?
11. Дан рекурсивный алгоритм F. Приведите последовательность чисел (без пробелов и разделителей), которые будут напечатаны на экране при выполнении вызова F(4).

Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач вывод n если n > 1 то F(n - 1) F(n - 2) все кон </pre>	<pre> procedure F(n: integer); begin write(n); if n > 1 then begin F(n - 1); F(n - 2) end end; </pre>
C++	Бейсик
<pre> void F(int n) { std::cout << n; if (n > 1) { F(n - 1); F(n - 2); } } </pre>	<pre> SUB F(n) PRINT n; IF n > 1 THEN F(n - 1) F(n - 2) END IF END SUB </pre>
Python	
<pre> def F(n): print(n) if n > 1: F(n - 1) F(n - 2) </pre>	

12. В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, - в виде четырех байтов, причем каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда – нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 231.117.87.213 адрес сети равен 231.117.87.208. Чему равно наибольшее возможное количество нулей в последнем байте маски сети?

13. При регистрации в компьютерной системе каждому пользователю выдается пароль, состоящий из 15 символов. Из соображений информационной безопасности каждый пароль должен содержать хотя бы 1 десятичную цифру, а также как прописные, так и строчные латинские буквы. 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 15 пользователях потребовалось 600 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответ запишите только целое число – количество байт.

14. Исполнитель Редактор получает на вход строку цифр и преобразовывает ее. Редактор может выполнять 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 . Например, выполнение команды

заменить (222, 34)

преобразует строку 77222277 в строку 7734277

Если в строке нет вхождений цепочки v , то выполнение команды **заменить (v, w)** не меняет эту строку.

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 исполнителя при этом не изменяется.

Цикл

ПОКА условие

 последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ условие

 ТО команда1

 ИНАЧЕ команда 2

КОНЕЦ ЕСЛИ

выполняется команда 1 (если условие истинно) или команда2 (если условие ложно).

Какая строка получится в результате применения приведенной ниже программы в строке, состоящей из 64 идущих подряд цифр 5? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (444) ИЛИ нашлось (555)

 ЕСЛИ нашлось (444)

 ТО заменить (444, 5)

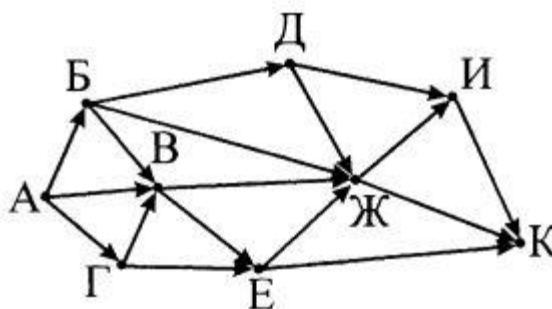
 ИНАЧЕ заменить (555, 4)

 КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

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



16. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 32 оканчивается на 4.
17. В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ «|», а для логической операции «И» - «&». В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

Запрос	Количество найденных страниц (в тысячах)
Колбаса	11
Паштет	12
Сыр	15
Сыр & Паштет	5
Колбаса & Сыр	4
Колбаса & Паштет	0

Какое количество страниц (в тысячах) будет найдено этой поисковой системой по запросу:

Колбаса | Сыр | Паштет ?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

18. Укажите наибольшее целое неотрицательное значение A , при котором выражение $(y < x) \vee (A < y) \vee (2x + 3y < 140)$ истинно для любых целых неотрицательных значений x и y .
19. В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 4; 5; 4; 7; 10; 7; 9; 11; 7; 8 соответственно, т.е. $A[0] = 4$; $A[1] = 5$ и т.д. Определите значение переменной k после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

Бейсик	Алгоритмический язык
<pre> k = 0 n = 3 FOR i = 0 TO 9 IF A(i) >= A(n) THEN k = k + A(i) MOD A(n) t = A(i) A(i) = A(n) A(n) = t END IF NEXT i </pre>	<pre> k := 0 n := 3 нц для i от 0 до 9 если A[i] >= A[n] то k := k + mod(A[i], A[n]) t := A[i] A[i] := A[n] A[n] := t все кц </pre>
C++	Паскаль
<pre> k = 0; n = 3; for (i = 0 ; i <= 9 ; i++) if (A[i] >= A[n]) { k = k + A[i] % A[n]; t = A[i]; A[i] = A[n]; A[n] = t; } </pre>	<pre> k := 0; n := 3; for i := 0 to 9 do if A[i] >= A[n] then begin k := k + A[i] mod A[n]; t := A[i]; A[i] := A[n]; A[n] := t end; end; </pre>
Python	
<pre> k = 0 n = 3 for i in range(10): if A[i] >= A[n]: k = k + A[i] % A[n] t = A[i] A[i] = A[n] A[n] = t </pre>	

20. Ниже на пяти языках записал алгоритм. Получив на вход число x , этот алгоритм печатает два числа: a и b . Укажите наименьшее из таких чисел x , при вводе которого алгоритм печатает сначала 3, а потом 4.

Бейсик	Паскаль
<pre> DIM X, A, B AS INTEGER INPUT X A = 0 : B = 0 WHILE X > 0 IF X MOD 2 > 0 THEN A = A + 1 ELSE B = B + 1 END IF X = X \ 2 WEND PRINT A PRINT B </pre>	<pre> var x, a, b: integer; begin readln(x); a := 0; b := 0; while x > 0 do begin if x mod 2 > 0 then a := a + 1 else b := b + 1; x := x div 2 end; writeln(a); write(b) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int x, a, b; cin >> x; a = 0; b = 0; while (x > 0) { if(x % 2 > 0) a = a + 1; else b = b + 1; x = x / 2; } cout << a << endl << b; return 0; } </pre>	<pre> алг нач цел x, a, b ввод x a := 0 b := 0 нц пока x > 0 если mod(x,2) > 0 то a := a + 1 иначе b := b + 1 все x := div(x,2) кц вывод а, нс, b кон </pre>
Python	
<pre> x = int(input()) a = 0; b = 0 while x > 0: if x % 2 > 0: a = a + 1 else: b = b + 1 x = x // 2 print(a) print(b) </pre>	

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

Бейсик	Паскаль
<pre> DIM A, B, T, M, R AS INTEGER A = -30: B = 30 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 = (x*x - 25)*(x*x - 25) + 8 END FUNCTION </pre>	<pre> var a,b,t,M,R : integer; function F(x:integer):integer; begin F := (x*x-25)*(x*x-25) + 8 end; BEGIN a := -30; b := 30; M := a; R := F(a); for t := a to b do if F(t) < R then begin M := t; R := F(t) end; write(M + R) END. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; long F(long x) { return (x*x-25)*(x*x-25) + 8; } int main() { long a, b, t, M, R; a = -30; b = 30; M = a; R = F(a); for (t = a; t <= b; t++) if (F(t) < R) { M = t; R = F(t); } cout << M + R << endl; return 0; } </pre>	<pre> алг нач цел a, b, t, M, R a := -30; b := 30 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 := (x*x-25)*(x*x-25) + 8 кон </pre>
Python	
<pre> def F(x): return (x*x - 25)*(x*x - 25) + 8 a = -30; b = 30 M = a; R = F(a) for t in range(a,b+1): if (F(t) < R): M = t; R = F(t) print(M + R) </pre>	

22. Исполнитель Увеличитель преобразует число на экране. У исполнителя есть три команды, которым присвоены номера:

1. Прибавить 1
2. Прибавить 2
3. Умножить на 3

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

Программа для исполнителя Увеличитель – это последовательность команд.

Сколько существует программ, для которых при исходном числе 2 результатом является число 20 и при этом траектория вычислений содержит число 11 и число 16?

Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы 123 при исходном числе 7 траектория будет состоять из чисел 8, 10, 30.

23. Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_7, x_8$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \wedge x_2) \vee \neg(x_3 \wedge x_4) = 1$$

$$(x_3 \wedge x_4) \vee \neg(x_5 \wedge x_6) = 1$$

$$(x_5 \wedge x_6) \vee \neg(x_7 \wedge x_8) = 1$$

В ответе **не нужно** перечислять все различные наборы значений $x_1, x_2, \dots, x_7, x_8$, при которых выполнена данная система неравенств. В качестве ответа вам нужно указать количество таких наборов.

Часть 2

24. Требовалось написать программу, которая вводит с клавиатуры натуральное число N ($N < 10^9$) и выводит на экран сумму нечетных цифр числа. Если в числе нет нечетных цифр, программа должна вывести на экран «NO». Ученик торопился и написал программу неправильно.

Паскаль	Бейсик
<pre> var N, s, d: integer; begin readln(N); s := 0; while N > 0 do begin d := N mod 10; if d mod 2 <> 0 then s := s + 1; N := N div 10 end; if s mod 2 = 0 then writeln('NO') else writeln(s) end. </pre>	<pre> DIM N AS LONG INPUT N s = 0 WHILE N > 0 d = N MOD 10 IF d MOD 2 <> 0 THEN s = s + 1 END IF N = N \ 10 WEND IF s MOD 2 = 0 THEN PRINT "NO" ELSE PRINT s END IF END </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { long int N; int d, s; cin >> N; s = 0; while (N > 0) { d = N % 10; if (d % 2 != 0) s = s + 1; N = N / 10; } if (s % 2 == 0) cout << "NO" << endl; else cout << s << endl; } </pre>	<pre> алг нач цел N, s, d ввод N s := 0 нц пока N > 0 d := mod(N, 10) если mod(d, 2) <> 0 то s := s + 1 все N := div(N, 10) кц если mod(s, 2) = 0 то вывод 'NO' иначе вывод s все кон </pre>
Python	
<pre> N = int(input()) s = 0 while N > 0: d = N % 10 if d % 2 != 0: s = s + 1 N = N // 10 if s % 2 == 0: print("NO") else: print(s) </pre>	

Выполните следующие действия:

- 1) Напишите, что выведет программа при вводе числа 328.
 - 2) Приведите такое наибольшее трехзначное число, в котором есть хотя бы одна нечетная цифра, при котором программа работает верно.
 - 3) Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки: выпишите строку, которая написана неправильно и приведите правильный вариант строки.
25. Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно.
- Опишите на одном из языков программирования алгоритм, который находит сумму нечетных элементов, не больших 99, а затем заменяет каждый такой элемент на число, равное найденной сумме. Гарантируется, что хотя бы один такой элемент в

массиве есть. В качестве результата необходимо вывести измененный массив, каждый элемент массива выводится с новой строки.

Например, для массива из 9-ти элементов: 427 99 300 52 1234 65 16 7 145

программа должна вывести числа 427 171 300 52 1234 171 16 171 145

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Паскаль	Бейсик
<pre>const N = 40; var a: array [1..N] of integer; i, s : integer; begin for i:=1 to N do read(a[i]); ... end.</pre>	<pre>N = 40 DIM A(N) AS INTEGER DIM I, S AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; const int N = 40; int main() { long a[N]; long i, s; for (i = 0; i < N; i++) cin >> a[i]; ... return 0; }</pre>	<pre>алг нач цел N = 40 целтаб a[1:N] цел i, s нц для i от 1 до N ввод a[i] кц ... кон</pre>
Python	
<pre># допускается также использовать # целочисленную переменную s a = [] n = 40 for i in range(0, n): a.append(int(input())) ...</pre>	

В качестве примера вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскаль).

26. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в кучу 2 камня или добавить в кучу 3 камня или увеличить количество камней в куче в два раза. Например, имея кучу из 8 камней, за один ход можно получить кучу из 10, 11 или 16 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 51. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 51 или больше камней.

В начальный момент в куче было S камней, $1 \leq S \leq 50$.

Говорят, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он

должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания:

- 1) а) При каких значениях числа S Петя может выиграть первым ходом? Укажите все такие значения и соответствующие выигрывающие ходы.
б) Укажите все такие значения S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Для каждого из указанных значений опишите выигрышную стратегию Вани.
 - 2) Укажите все значения S , при которых у Пети есть выигрышная стратегия, причем Петя не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанных значений S опишите выигрышную стратегию Пети.
 - 3) Укажите такое значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, но при этом у Вани нет стратегии, которая позволяет ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На ребрах указывайте, кто делает ход, в узлах – количество камней в куче.
27. На вход программы подается: в первой строке – число N ($5 < N < 10^4$).
В каждой из последующих N строк – по одному элементу последовательности – целые положительные числа, не превышающие 10^9 .

Напишите эффективную, в том числе и по используемой памяти, программу, которая находит в последовательности такую пару элементов, у которой одновременно выполняются следующие условия:

- максимальная сумма,
- номера элементов различаются не меньше чем на 5,
- сумма должна быть кратна m ($m = 20$),
- элемент пары, который появился в последовательности первым, должен быть больше второго элемента пары.

В качестве ответа необходимо вывести на экран два найденных элемента пары. Гарантируется, что в последовательности всегда найдется такая пара.

Укажите используемую версию языка программирования, например, Free Pascal 2.6. Программу будем называть эффективной по времени, если при увеличении количества элементов N в k раз и при увеличении параметра m в k раз время работы программы увеличится не более чем в k раз.

Программу будем называть эффективной по памяти, если объем данных, используемый для работы программы, не зависит от количества входных данных N и не превышает 1 килобайта.

Вы можете написать две программы. Итоговый балл за эту задачу вы получите как больший из баллов за эти программы.

Пример входных данных:

8
13
25
4
86
8
3
7

15

Пример выходных данных:

25 15

V. Ответы на тесты по информатике и ИКТ

	Примерный вариант
1	3
2	ywzx
3	25
4	689
5	100
6	20
7	10
8	88
9	32
10	5
11	432101210
12	4
13	28
14	4455
15	23
16	7,14,28
17	29
18	27
19	4
20	67
21	3
22	2600
23	121

Задание 24

1. 1
2. 881
- 3.

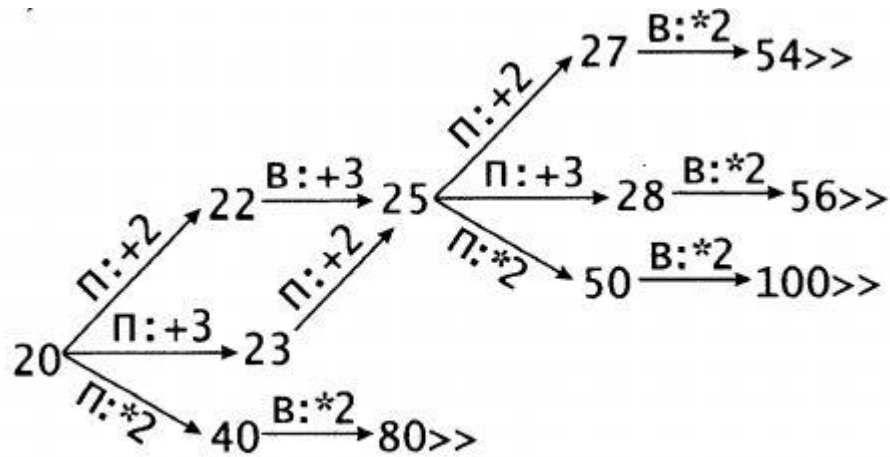
Неверно	Верно
<code>s := s + 1;</code>	<code>s := s + d;</code>
<code>if s mod 2 = 0 then</code>	<code>if s = 0 then</code>

Задание 25

Паскаль	Бейсик
<pre> s := 0; for i := 1 to N do if (a[i] mod 2 <> 0) and (a[i] <= 99) then s := s + a[i]; for i := 1 to N do if (a[i] mod 2 <> 0) and (a[i] <= 99) then a[i] := s; for i := 1 to N do writeln(a[i]) </pre>	<pre> S = 0 FOR I = 1 TO N IF A(I) MOD 2 <> 0 AND A(I) <= 99 THEN S = S + A(I) END IF NEXT I FOR I = 1 TO N IF A(I) MOD 2 <> 0 AND A(I) <= 99 THEN A(I) = S END IF NEXT I FOR I = 1 TO N PRINT A(I) NEXT I </pre>
C++	Алгоритмический язык
<pre> s = 0; for (i = 0 ; i < N ; i++) if (a[i] % 2 != 0 && a[i] <= 99) s += a[i]; for (i = 0 ; i < N ; i++) if (a[i] % 2 != 0 && a[i] <= 99) a[i] = s; for (i = 0 ; i < N ; i++) cout << a[i] << endl; </pre>	<pre> s := 0 нц для i от 1 до N если mod(a[i],2)<>0 и a[i] <= 99 то s := s + a[i] все кц нц для i от 1 до N если mod(a[i],2)<>0 и a[i] <= 99 то a[i] := s все кц нц для i от 1 до N вывод a[i] кц </pre>
Python	
<pre> s = 0 for i in range (0, n): if a[i] % 2 != 0 and a[i] <= 99: s = s + a[i] for i in range (0, n): if a[i] % 2 != 0 and a[i] <= 99: a[i] = s for i in range (0, n): print(a[i]) </pre>	

Задание 26

- 1) а) $S = 26, \dots, 50$.
- б) $S = 24, S = 25$.
- 2) $S = 12, S = 21, S = 22, S = 23$.
- 3) $S = 20$



Задание 27

Затем будем сдвигать буфер влево и дописывать в его конец текущий элемент.
Напишем программу на PascalABC.

```

const m = 20;
var b : array[1..5] of integer; // буфер
    r : array[0..m-1] of integer; // наибольшие числа для каждого остатка
    a, i, j, m1, m2, p, q, n : integer;
begin
    read(n);
    for i := 1 to 5 do // считываем первые 5 элементов в буфер
        read(b[i]);
    for i := 0 to m-1 do // обнуляем максимальные числа для каждого остатка
        r[i] := 0;
    m1 := 0; m2 := 0; // наибольшие искомые элементы пары
    for i := 6 to n do // перебираем все остальные числа последовательности
        begin
            // обрабатываем первый элемент буфера
            q := b[1] mod m; // вычисляем его остаток от деления на m
            if b[1] > r[q] then // если b[1] больше текущего максимума
                // с таким остатком
                r[q] := b[1]; // то обновляем этот максимум
            read(a); // считываем очередной элемент последовательности
            p := m - a mod m; // вычисляем «дополнение» к его остатку
            if p = m then // если оно получилось равно m, то
                p := 0; // обнуляем это дополнение
            // если левый элемент пары больше правого и
            // если их сумма больше текущего максимума
            if (r[p] > a) and (r[p] + a > m1 + m2) then
                begin
                    m1 := r[p]; // обновляем текущую максимальную пару
                    m2 := a
                end;
            for j := 1 to 4 do // сдвигаем буфер влево
                b[j] := b[j+1];
            b[5] := a // кладем в его конец текущий элемент
        end;
        write(m1, ' ', m2)
    end.
  
```