

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ
Государственное автономное образовательное учреждение
Высшего образования города Москвы
«Московский городской педагогический университет»
(ГАОУ ВО МГПУ)



Программа вступительного испытания
«Информатика и информационно-коммуникационные технологии»
(Бакалавриат)

Разработчики программы:

1. А.В. Нестеров, доктор физико-математических наук, профессор, профессор кафедры прикладной информатики
2. В.Ш. Крупник, кандидат технических наук, доцент, доцент кафедры прикладной информатики
3. Л.А. Пономарева, кандидат физико-математических наук, доцент, доцент кафедры прикладной информатики

СОДЕРЖАНИЕ ПРОГРАММЫ

1. Пояснительная записка [подробнее ➤](#)
2. Критерии оценки [подробнее ➤](#)
3. Программа вступительного испытания [подробнее ➤](#)
4. Примерные экзаменационные задания [подробнее ➤](#)
5. Список рекомендованной литературы [подробнее ➤](#)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа вступительного испытания составлена на основе федерального государственного образовательного стандарта среднего общего образования и федерального государственного образовательного стандарта основного общего образования.

Экзамен проводится в письменной форме.

Вступительное испытание включает в себя вопросы в форме тестов. Абитуриентам предлагается решить 22 задачи, сведенных в три группы заданий.

На каждую из первых шестнадцати задач экзаменуемым будет предложено четыре варианта ответа, помеченные цифрами от (1) до (4), из которых требуется выбрать один.

Следующие четыре задачи абитуриент должен будет решить и выписать ответ.

Каждую из двух последних задач абитуриент должен будет решить, написать ответ и изложить в письменном виде решение задачи, как это делается в обычной контрольной работе. Способ решения и выбор формы изложения решения не влияют на оценку задачи с единственной оговоркой: решение должно быть изложено в форме, понятной для экзаменатора.

Продолжительность экзамена составляет 3 часа 55 минут (235 минут).

Поступающий допускается к сдаче вступительного испытания при наличии у него паспорта или иного документа, удостоверяющего его личность.

При опоздании к началу вступительного испытания поступающий может быть допущен к испытанию, причем время на выполнение задания ему не увеличивается.

Лица, не явившиеся на вступительные испытания по уважительной причине (болезнь или иные обстоятельства, подтвержденные документально), допускаются к ним в резервный день.

Во время проведения вступительных испытаний их участникам и лицам, привлекаемым к их проведению, запрещается иметь при себе и использовать средства связи (мобильные телефоны, планшеты и т.п.).

При несоблюдении поступающим порядка проведения вступительных испытаний, экзаменационные комиссии, проводящие вступительное испытание

вправе удалить поступающего с места проведения вступительного испытания с составлением акта об удалении. В случае удаления поступающего со вступительного испытания Университет возвращает поступающему принятые документы и не допускает до участия в конкурсе.

КРИТЕРИИ ОЦЕНКИ

Максимальная оценка за экзамен – 100 баллов.

Предлагаемые первые шестнадцать задач оцениваются в 3 балла каждая. Если абитуриент не даёт ответа на задачу или даёт неправильный ответ, то он получает за него 0 баллов. Таким образом, общая сумма баллов, которую может набрать абитуриент за решение первых шестнадцати задач, составляет 48 баллов.

Ответ на следующие четыре задачи будет оценен в 7 баллов.

Правильное решение 21-й и 22-й задач из этой группы будет оценено в 12 и 12 баллов соответственно. Способ решения и выбор формы изложения решения не влияют на оценку задачи с единственной оговоркой: решение должно быть изложено в форме, понятной для экзаменатора.

Баллы, полученные за все выполненные задания, суммируются.

Абитуриент, набравший по итогам экзамена, ниже установленного Университетом минимального балла, считается не сдавшим вступительное испытание и выбывает из участия в конкурсе.

ПРОГРАММА ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ

Вступительное испытание (экзамен) по дисциплине «Информатика и ИКТ» включает в себя вопросы в форме тестов. Абитуриентам предлагается решить 22 задачи, сведенных в три группы заданий.

Темы, которые должен знать абитуриент

Тема: Системы счисления и двоичное представление информации в памяти компьютера.

Что нужно знать:

- перевод чисел между десятичной, двоичной, восьмеричной и шестнадцатеричной системами счисления

Полезно помнить, что в двоичной системе:

- четные числа оканчиваются на 0, нечетные – на 1;
- числа, которые делятся на 4, оканчиваются на 00, и т.д.; числа, которые делятся на 2^k , оканчиваются на k нулей
- если число N принадлежит интервалу $2^{k-1} \leq N < 2^k$, в его двоичной записи будет всего k цифр, например, для числа **125**:

$$2^6 = 64 \leq \mathbf{125} < 128 = 2^7, \quad 125 = 1111101_2 \text{ (7 цифр)}$$

- числа вида 2^k записываются в двоичной системе как единица и k нулей, например:

$$16 = 2^4 = 10000_2$$

- числа вида $2^k - 1$ записываются в двоичной системе k единиц, например:

$$15 = 2^4 - 1 = 1111_2$$

- если известна двоичная запись числа N , то двоичную запись числа $2 \cdot N$ можно легко получить, приписав в конец ноль, например:

$$\bullet \quad 15 = 1111_2, \quad 30 = 11110_2, \quad 60 = 111100_2, \quad 120 = 1111000_2$$

- отрицательные целые числа хранятся в памяти в двоичном дополнительном коде.

Тема: Вычисление информационного объема сообщения.

Что нужно знать:

- с помощью K бит можно закодировать $Q = 2^K$ различных вариантов (чисел)
- таблица степеней двойки, она же показывает, сколько вариантов Q можно

закодировать с помощью K бит:

K , бит	1	2	3	4	5	6	7	8	9	10
Q , вариантов	2	4	8	16	32	64	128	256	512	1024

- при измерении количества информации принимается, что в одном байте 8 бит, а в одном килобайте (1 кбайт) – 1024 байта, в мегабайте (1Мбайт) – 1024 кбайта;

- чтобы найти информационный объем сообщения (текста) I , нужно умножить количество символов (отсчетов) N на число бит на символ (отсчет) K : $I = N * K$;

- две строчки текста не могут занимать 100 кбайт в памяти;
- мощность алфавита – это количество символов в этом алфавите;
- для перевода отрицательного числа (**-a**) в двоичный дополнительный код

нужно сделать следующие операции;

- перевести число **a-1** в двоичную систему счисления:

сделать инверсию битов: заменить все нули на единицы и единицы на нули в пределах разрядной сетки.

Тема: Выполнение арифметических операций в двоичной, восьмеричной и шестнадцатеричной системах счисления.

Что нужно знать:

- перевод чисел между десятичной, двоичной, восьмеричной и шестнадцатеричной системами счисления выполнение сложения (вычитания, умножения) в этих системах.

Тема: Оператор присваивания в языке программирования.

Что нужно знать:

- переменная – это величина, которая имеет имя, тип и значение; переменная может изменяться во время выполнения программы;
- оператор присваивания служит для записи значения в переменную;
- если в переменную записывают новое значение, старое стирается;
- знаки +, −, *, / используются для обозначения операций сложения, вычитания, умножения и деления;
- запись вида **a div b** означает результат целочисленного деления **a** на **b** (остаток отбрасывается);
- запись вида **a mod b** означает остаток от деления **a** на **b**;
- запись вида **a := b + 2*c + 3;** означает «вычислить значения выражения справа от знака присваивания := и записать результат в переменную **a**»; при этом значения других переменных (кроме **a**) не изменяются;
- для многократного выполнения одинаковых операций используют циклы;
- цикл с переменной выполняется **N** раз, в этом примере переменная **i** принимает последовательно все значения от 1 до **N** с шагом 1

for i:=1 to N do begin

{ что-то делаем }

end;

- цикл с условием выполняется до тех пор, пока условие в заголовке цикла не нарушится;

while { условие } do begin

{ что-то делаем }

end;

главная опасность при использовании цикла с условием – **зацикливание**; эта такая ситуация, когда цикл работает бесконечно долго из-за того, что условие все время остается.

Тема: Работа с массивами и матрицами в языке программирования.

Что нужно знать:

- работу цикла **for** (цикла с переменной);
- массив – это набор однотипных элементов, имеющих общее имя и расположенных в памяти рядом;
 - для обращения к элементу массива используют квадратные скобки, запись **A[i]** обозначает элемент массива **A** с номером (индексом) **i**;
 - матрица (двухмерный массив) – это прямоугольная таблица однотипных элементов;
 - если матрица имеет имя **A**, то обращение **A[i,k]** обозначает элемент, расположенный на пересечении строки **i** и столбца **k**;
 - элементы, у которых номера строки и столбца совпадают, расположены на главной диагонали

A[1,1]			
	A[2,2]		
		A[3,3]	
			A[4,4]

- выше главной диагонали расположены элементы, у которых номер строки **меньше** номера столбца:

	A[1,2]	A[1,3]	A[1,4]
		A[2,3]	A[2,4]
			A[3,4]

- ниже главной диагонали расположены элементы, у которых номер строки **больше** номера столбца:

A[2,1]			

A[3,1]	A[3,2]		
A[4,1]	A[4,2]	A[4,3]	

Тема: Основные понятия математической логики.

Что нужно знать:

- условные обозначения логических операций

$\neg A, \bar{A}$ не A (отрицание, инверсия)

$A \wedge B, A \cdot B$ A и B (логическое умножение, конъюнкция)

$A \vee B, A + B$ A или B (логическое сложение, дизъюнкция)

$A \rightarrow B$ импликация (следование)

- таблицы истинности логических операций «И», «ИЛИ», «НЕ», «импликация»

- операцию «импликация» можно выразить через «ИЛИ» и «НЕ»:

$A \rightarrow B = \neg A \vee B$ или в других обозначениях $A \rightarrow B = \bar{A} + B$

- если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем – «ИЛИ», и самая последняя – «импликация»

- иногда полезны формулы де Моргана¹:

$$\neg (A \wedge B) = \neg A \vee \neg B \quad \overline{A \cdot B} = \bar{A} + \bar{B}$$

$$\neg (A \vee B) = \neg A \wedge \neg B \quad \overline{A + B} = \bar{A} \cdot \bar{B}$$

Тема: Преобразование логических выражений. Формулы де Моргана.

Что нужно знать:

- условные обозначения логических операций

$\neg A, \bar{A}$ не A (отрицание, инверсия)

$A \wedge B, A \cdot B$ A и B (логическое умножение, конъюнкция)

$A \vee B, A + B$ A или B (логическое сложение, дизъюнкция)

$A \rightarrow B$ импликация (следование)

¹ Огастес (Август) де Морган – шотландский математик и логик.

- операцию «импликация» можно выразить через «ИЛИ» и «НЕ»:

$$A \rightarrow B = \neg A \vee B \text{ или в других обозначениях } A \rightarrow B = \bar{A} + B$$

• если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем – «ИЛИ», и самая последняя – «импликация»

- правила преобразования логических выражений

Законы алгебры логики

21

название	для И	для ИЛИ
двойного отрицания	$\overline{\bar{A}} = A$	
исключения третьего	$A \cdot \bar{A} = 0$	$A + \bar{A} = 1$
операции с константами	$A \cdot 0 = 0, A \cdot 1 = A$	$A + 0 = A, A + 1 = 1$
повторения	$A \cdot A = A$	$A + A = A$
поглощения	$A \cdot (A + B) = A$	$A + A \cdot B = A$
переместительный	$A \cdot B = B \cdot A$	$A + B = B + A$
сочетательный	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$
распределительный	$A + B \cdot C = (A + B) \cdot (A + C)$	$A \cdot (B + C) = A \cdot B + A \cdot C$
законы де Моргана	$\overline{A \cdot B} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A} \cdot \bar{B}$

• фактически это задание на применение законов де Моргана (хотя об этом нигде не говорится):

$$\neg (A \wedge B) = \neg A \vee \neg B \qquad \overline{A \cdot B} = \bar{A} + \bar{B}$$

$$\neg (A \vee B) = \neg A \wedge \neg B \qquad \overline{A + B} = \bar{A} \cdot \bar{B}$$

Тема: Построение таблиц истинности логических выражений.

Что нужно знать:

- условные обозначения логических операций

$\neg A, \bar{A}$ не A (отрицание, инверсия)

$A \wedge B, A \cdot B$ A и B (логическое умножение, конъюнкция)

$A \vee B, A + B$ A или B (логическое сложение, дизъюнкция)

$A \rightarrow B$ импликация (следование)

- операцию «импликация» можно выразить через «ИЛИ» и «НЕ»:

$A \rightarrow B = \neg A \vee B$ или в других обозначениях $A \rightarrow B = \bar{A} + B$

- иногда для упрощения выражений полезны формулы де Моргана:

$$\neg (A \wedge B) = \neg A \vee \neg B \qquad \overline{A \cdot B} = \bar{A} + \bar{B}$$

$$\neg (A \vee B) = \neg A \wedge \neg B \qquad \overline{A + B} = \bar{A} \cdot \bar{B}$$

- если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем – «ИЛИ», и самая последняя – «импликация»;

- таблица истинности выражения определяет его значения при всех возможных комбинациях исходных данных;

- если известна только часть таблицы истинности, соответствующее логическое выражение однозначно определить нельзя, поскольку частичной таблице могут соответствовать несколько *разных* логических выражений (не совпадающих для других вариантов входных данных);

- количество *разных* логических выражений, удовлетворяющих неполной таблице истинности, равно 2^k , где k – число *отсутствующих* строк; например, полная таблица истинности выражения с тремя переменными содержит $2^3=8$ строчек, если заданы только 6 из них, то можно найти $2^{8-6}=2^2=4$ *разных* логических выражения, удовлетворяющие этим 6 строчкам (но отличающиеся в двух оставшихся);

- логическая сумма $A + B + C + \dots$ равна 0 (выражение ложно) тогда и только тогда, когда все слагаемые одновременно равны нулю, а в остальных случаях равна 1 (выражение истинно);

- логическое произведение $A \cdot B \cdot C \cdot \dots$ равно 1 (выражение истинно) тогда и только тогда, когда все сомножители одновременно равны единице, а в остальных случаях равно 0 (выражение ложно).

Тема: Использование информационных моделей (таблицы, диаграммы, графики). Перебор вариантов, выбор лучшего по какому-то признаку.

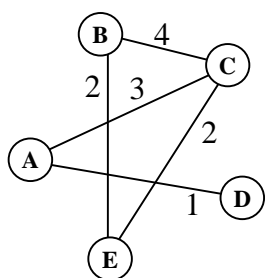
Что нужно знать:

- умение перебирать варианты (не пропустив ни одного!);

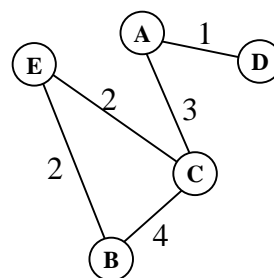
- полезно знать, что такое *граф* (это набор вершин и соединяющих их ребер) и как он описывается в виде таблицы, хотя, как правило, все необходимые объяснения даны в формулировке задания;

- чаще всего используется *взвешенный граф*, где с каждым ребром связано некоторое число (вес), оно может обозначать, например, расстояние между городами или стоимость перевозки;

- рассмотрим граф (рисунок слева), в котором 5 вершин (А, В, С, D и E); он описывается таблицей, расположенной в центре; в ней, например, число 4 на пересечении строки В и столбца С означает, что, во-первых, есть ребро, соединяющее В и С, и во-вторых, вес этого ребра равен 4; пустая клетка на пересечении строки А и столбца В означает, что ребра из А в В нет



	A	B	C	D	E
A			3	1	
B			4		2
C	3	4			2
D	1				
E		2	2		



- обратите внимание, что граф по заданной таблице (она еще называется *весовой матрицей*) может быть нарисован по-разному; например, той же таблице соответствует граф, показанный на рисунке справа от нее;

- в приведенном примере матрица симметрична относительно главной диагонали; это может означать, например, что стоимости перевозки из В в С и обратно равны (это не всегда так);

- желательно научиться быстро (и правильно) строить граф по весовой матрице и наоборот.

Тема: Кодирование и декодирование информации.

Что нужно знать:

- кодирование – это перевод информации с одного языка на другой (запись в другой системе символов, в другом алфавите). Обычно кодированием называют перевод информации с «человеческого» языка на формальный, например, в двоичный код, а декодированием – обратный переход;
- один символ исходного сообщения может заменяться одним символом нового кода или несколькими символами, а может быть и наоборот – несколько символов исходного сообщения заменяются одним символом в новом коде (китайские иероглифы обозначают целые слова и понятия);
- кодирование может быть *равномерное* и *неравномерное*; при равномерном кодировании все символы кодируются кодами равной длины; при неравномерном кодировании разные символы могут кодироваться кодами разной длины, это затрудняет декодирование.

Тема: Проверка закономерностей методом рассуждений.

Что нужно знать:

- основы теории логики.

Тема: Файловая система.

Что нужно знать:

- данные на дисках хранятся в виде файлов (наборов данных, имеющих имя);
- чтобы было удобнее разбираться с множеством файлов, их объединяют в каталоги (в *Windows* каталоги называются «папками»);
- каталог можно воспринимать как контейнер, в котором размещаются файлы и другие каталоги, которые называются подкаталогами или *вложенными* каталогами (они находятся внутри другого каталога, вложены в него);
- каталоги организованы в многоуровневую (иерархическую) структуру, которая называется «деревом каталогов»;

- главный каталог диска (который пользователь видит, «открыв» диск, например, в Проводнике *Windows* или аналогичной программе) называется *корневым* каталогом или «корнем» диска, он обозначается буквой логического диска, за которой следует двоеточие и знак «\» (обратный слэш); например, **A:** – это обозначение корневого каталога диска A;

- каждый каталог (кроме корневого) имеет (один единственный!) «родительский» каталог – этот тот каталог, внутри которого находится данный каталог;

- полный адрес каталога – это перечисление каталогов, в которые нужно войти, чтобы попасть в этот каталог (начиная с корневого каталога диска); например, **C:\USER\BIN\SCHOOL;**

- полный адрес файла состоит из адреса каталога, в котором он находится, символа «\» и имени файла, например, **C:\USER\BIN\SCHOOL\Вася.txt;**

- маска служит для обозначения (выделения) группы файлов, имена которых имеют общие свойства, например, общее расширение;

- в масках, кроме «обычных» символов (допустимых в именах файлов) используются два специальных символа: звездочка «*» и знак вопроса «?»;

- звездочка «*» обозначает любое количество любых символов, в том числе, может обозначать пустую последовательность;

- знак вопроса «?» обозначает ровно один любой символ;

- при выводе списка имен файлов они могут быть отсортированы по имени, типу (расширению), дате последнего изменения, размеру; это не меняет их размещения на диске;

- если установлена сортировка по имени или типу, сравнение идет по кодам символов, входящих в имя или в расширение.

Тема: Поиск и сортировка информации в базах данных.

Что нужно знать:

- при составлении условия отбора можно использовать знаки отношений <, <= (меньше или равно), >, >= (больше или равно), = (равно), <> (не равно);

- последовательность выполнения логических операций в сложных запросах: сначала выполняются отношения, затем – «И», потом – «ИЛИ»;
- для изменения порядка выполнения операции используют скобки.

Тема: Кодирование и обработка графической информации.

Что нужно знать:

- графическая информация может храниться в растровом и векторном форматах;
- *векторное* изображение – это набор геометрических фигур, которые можно описать математическими зависимостями;
- *растровое* изображение хранится в виде набора пикселей, для каждого из которых задается свой цвет, независимо от других;
- *глубина цвета* – это количество бит на пиксель (обычно от 1 до 24 бит на пиксель);
- в режиме истинного цвета (*True Color*) информация о цвете каждого пикселя растрового изображения хранится в виде набора его RGB-составляющих (*Red, Green, Blue*); каждая из RGB-составляющих – целое число (яркость) в интервале $[0, 255]$ (всего 256 вариантов), занимающее в памяти 1 байт или 8 бит (так как $2^8 = 256$); таким образом, на каждый пиксель отводится 3 байта = 24 бита памяти (глубина цвета – 24 бита); нулевое значение какой-то составляющей означает, что ее нет в этом цвете, значение 255 – максимальная яркость; в режиме истинного цвета можно закодировать $256^3 = 2^{24} = 16\,777\,216$ различных цветов;
- *палитра* – это ограниченный набор цветов, которые используются в изображении (обычно не более 256); при кодировании с палитрой выбираются N любых цветов (из полного набора 16 777 216 цветов), для каждого из них определяется RGB-код и уникальный номер от 0 до $N-1$; тогда информация о цвете пикселя – это *номер* его цвета в палитре; при кодировании с палитрой количество бит на 1 пиксель (K) зависит от количества цветов в палитре N , они связаны формулой:

$N = 2^K$; объем памяти на все изображение вычисляется по формуле $M = Q \cdot K$, где K – число бит на пиксель, а Q – общее количество пикселей;

- полезно знать на память таблицу степеней двойки: она показывает, сколько вариантов N (а данном случае – сколько цветов) можно закодировать с помощью K бит:

К, бит	1	2	3	4	5	6	7	8	9	10
N, вариантов	2	4	8	16	32	64	128	256	512	1024

- цвет на Web-страницах кодируется в виде RGB-кода в шестнадцатеричной системе: #RRGGBB, где RR, GG и BB – яркости красного, зеленого и синего, записанные в виде двух шестнадцатеричных цифр; это позволяет закодировать 256 значений от 0 (00_{16}) до 255 (FF_{16}) для каждой составляющей; коды некоторых цветов: #FFFFFF – белый, #000000 – черный, #CCCCCC и любой цвет, где $R = G = B$, – это серый разных яркостей #FF0000 – красный, #00FF00 – зеленый, #0000FF – синий, #FFFF00 – желтый, #FF00FF – фиолетовый, #00FFFF – цвет морской волны

Тема: Электронные таблицы.

Что нужно знать:

- адрес ячейки в электронных таблицах состоит из имени столбца и следующего за ним номера строки, например, C15;
- формулы в электронных таблицах начинаются знаком = («равно»);
- знаки +, –, *, / и ^ в формулах означают соответственно сложение, вычитание, умножение, деление и возведение в степень;
- запись B2:C4 означает диапазон, то есть, все ячейки внутри прямоугольника, ограниченного ячейками B2 и C4:

	A	B	C	D
1				
2				
3				
4				
5				
6				

- например, по формуле =СУММ(B2:C4) вычисляется сумма значений ячеек B2, B3, B4, C2, C3 и C4;
- в заданиях могут использоваться стандартные функции СЧЕТ (количество непустых ячеек), СУММ (сумма), СРЗНАЧ (среднее значение), МИН (минимальное значение), МАКС (максимальное значение);
- функция СРЗНАЧ при вычислении среднего арифметического не учитывает пустые ячейки; например, после ввода формулы в C2 появится значение 2 (ячейка A2 – пустая):

	A	B	C
1	1	2	
2		3	
3			=СРЗНАЧ(A1:B2)

функция СЧЕТ(A1:B2) в этом случае выдаст значение 3 (а не 4).

- адреса ячеек (или ссылки на ячейки) бывают относительные, абсолютные и смешанные, вся разница между ними проявляется при копировании формулы в другую ячейку;
- в *абсолютных* адресах перед именем столбца и перед номером строки ставится знак доллара \$, такие адреса не изменяются при копировании; вот что будет, если формулу =**\$B\$2+\$C\$3** скопировать из D5 во все соседние ячейки

	C	D	E
4	=B\$2+\$C\$3	=B\$2+\$C\$3	=B\$2+\$C\$3
5	=B\$2+\$C\$3	=B\$2+\$C\$3	=B\$2+\$C\$3
6	=B\$2+\$C\$3	=B\$2+\$C\$3	=B\$2+\$C\$3

знак \$ как бы «фиксирует» значение: в абсолютных адресах и имя столбца, и номер строки зафиксированы;

- в *относительных* адресах знаков доллара нет, такие адреса при копировании изменяются: номер столбца (строки) изменяется на столько, на сколько отличается номер столбца (строки), где оказалась скопированная формула, от номера

столбца (строки) исходной ячейки; вот что будет, если формулу **=B2+C3** (в ней оба адреса – относительные) скопировать из D5 во все соседние ячейки:

	C	D	E
4	=A1+B2	=B1+C2	=C1+D2
5	=A2+B3	=B2+C3	=C2+D3
6	=A3+B4	=B3+C4	=C3+D4

- в *смешанных* адресах часть адреса (строка или столбец) – абсолютная, она «зафиксирована» знаком \$, а вторая часть – относительная; относительная часть изменится при копировании так же, как и для относительной ссылки:

	C	D	E
4	=\$B1+B\$3	=\$B1+C\$3	=\$B1+D\$3
5	=\$B2+B\$3	=\$B2+C\$3	=\$B2+D\$3
6	=\$B3+B\$3	=\$B3+C\$3	=\$B3+D\$3
7			

Тема: Представление данных в электронных таблицах в виде диаграмм и графиков.

Что нужно знать:

- что такое столбчатая, линейчатая и круговая диаграмма, какую информацию можно получить с каждой из них;
- адрес ячейки в электронных таблицах состоит из имени столбца и следующего за ним номера строки, например, C15;
- формулы в электронных таблицах начинаются знаком = («равно»);
- знаки +, -, *, / и ^ в формулах означают соответственно сложение, вычитание, умножение, деление и возведение в степень
- в заданиях могут использоваться стандартные функции СУММ (сумма), СРЗНАЧ (среднее значение), МИН (минимальное значение), МАКС (максимальное значение);
- запись B2:C4 означает диапазон, то есть, все ячейки внутри прямоугольника, ограниченного ячейками B2 и C4; например, с помощью формулы =СУММ(B2:C4) вычисляется сумма значений ячеек B2, B3, B4, C2, C3 и C4.

Тема: Выполнение алгоритмов для исполнителя.

Что нужно знать:

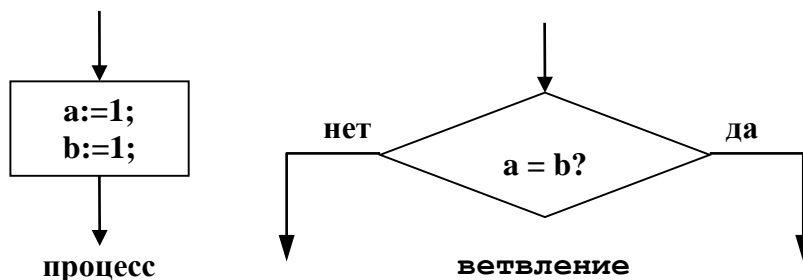
- правила выполнения линейных, разветвляющихся и циклических алгоритмов;
- основные операции с символьными строками (определение длины, выделение подстроки, удаление и вставка символов, «сцепка» двух строк в одну);
- *исполнитель* – это человек, группа людей, животное, машина или другой объект, который может понимать и выполнять некоторые команды;
- в школьном алгоритмическом языке **нц** обозначает «начало цикла», а **кц** – «конец цикла»; все команды между **нц** и **кц** – это тело цикла, они выполняются несколько раз;
- запись **нц для i от 1 до n** обозначает начало цикла, в котором переменная **i** (она называется переменной цикла) принимает последовательно все значения от 1 до **n** с шагом 1.

Тема: Блок-схемы алгоритмов. Переменные, присваивание значений. Ветвления. Организация циклов с помощью блока «ветвление».

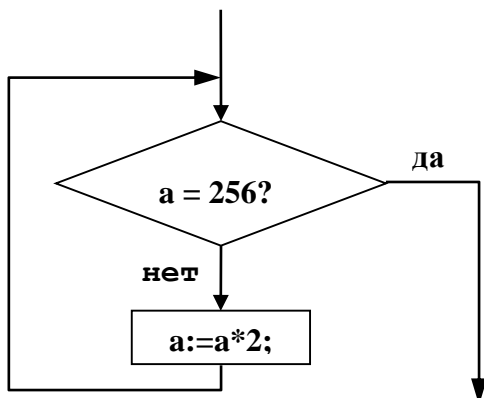
Что нужно знать:

- переменная – это величина, которая имеет имя, тип и значение; переменная может изменяться во время выполнения программы;
- оператор присваивания (в Паскале обозначается сочетанием символов «:=») служит для записи нового значения в переменную (для изменения ее значения);
- если в переменную записывают новое значение, старое стирается;
- знаки +, -, *, / используются для обозначения операций сложения, вычитания, умножения и деления;
- запись вида **a := a + 2;** – это не уравнение, а команда «прочитать текущее значение переменной **a**, добавить к нему 2 и записать результат обратно в переменную **a**»;
- для наглядной записи небольших алгоритмов используют блок-схемы; они состоят из блоков разного назначения и соединительных линий со стрелками, которые показывают порядок выполнения блоков

- в задачах встречаются два блока: *процесс* (выполнение некоторых действий) и *ветвление* (условие, в зависимости от которого выполнение алгоритма продолжается по одной или другой «ветке»)



- с помощью ветвления можно организовать *цикл* (многократное выполнение одинаковых действий), в этом случае в блок-схеме будет соединительная линия, идущая «в обратном направлении» (петля, замкнутый контур);
- цикл на рисунке (выделен зеленым фоном) закончится только тогда, когда выполнится условие **a = 256**



Тема: Кодирование чисел. Системы счисления.

Что нужно знать:

- принципы кодирования чисел в позиционных системах счисления;
- чтобы перевести число, скажем, 12345_N , из системы счисления с основанием N в десятичную систему, нужно умножить значение каждой цифры на N в степени, равной ее разряду:

4 3 2 1 0 ← разряды

$$1\ 2\ 3\ 4\ 5_N = 1 \cdot N^4 + 2 \cdot N^3 + 3 \cdot N^2 + 4 \cdot N^1 + 5 \cdot N^0$$

- последняя цифра записи числа в системе счисления с основанием N – это остаток от деления этого числа на N ;
- две последние цифры – это остаток от деления на N^2 , и т.д.

Тема: Поиск алгоритма минимальной длины для исполнителя.

Что нужно знать:

- умение проводить простой перебор вариантов, организованный оптимальным образом;
- *исполнитель* – это человек, группа людей, животное, машина или другой объект, который может понимать и выполнять некоторые команды.

Тема: Определение скорости передачи информации при заданной пропускной способности канала.

Что нужно знать:

- «физический» аналог задачи:



пропускная способность – 10

сколько лимонада перекачается по трубе за 1 час?

ответ: $10 \text{ л/мин} \cdot 60 \text{ мин} = 600 \text{ л}$

- любой канал связи имеет ограниченную пропускную способность (скорость передачи информации), это число ограничивается свойствами аппаратуры и самой линии (кабеля);
- объем переданной информации Q вычисляется по формуле $Q = q \cdot t$, где q – пропускная способность канала (в битах в секунду или подобных единицах), а t – время передачи.

Тема: Анализ алгоритма построения последовательности.

Что нужно знать:

- в некоторых задачах (на RLE-кодирование, см. далее) нужно знать, что такое бит и байт, что байт равен 8 бит, что такое старший бит, как переводить числа из двоичной системы в десятичную;
- умение решать классические задачи на символьные цепочки.

Тема: Компьютерные сети. Адресация в Интернете.

Что нужно знать:

- каждый компьютер, подключенный к сети Интернет, должен иметь собственный адрес, который называют IP-адресом (IP = *Internet Protocol*);
- IP-адрес состоит из четырех чисел, разделенных точками; каждое из этих чисел находится в интервале 0...255, например: **192.168.85.210**;
- адрес документа в Интернете (URL = *Uniform Resource Locator*) состоит из следующих частей:
 - протокол, чаще всего **http** (для Web-страниц) или **ftp** (для файловых архивов)
 - знаки **://**, отделяющие протокол от остальной части адреса
 - доменное имя (или IP-адрес) сайта
 - каталог на сервере, где находится файл
 - имя файла
- принято разделять каталоги не обратным слэшем «\» (как в *Windows*), а прямым «/», как в системе *UNIX* и ее «родственников», например, в *Linux*;
- пример адреса (URL)

<http://www.vasya.ru/home/user/vasya/qu-qu.zip>

здесь желтым маркером выделен протокол, фиолетовым – доменное имя сайта, голубым – каталог на сайте и серым – имя файла.

Тема: Составление запросов для поисковых систем с использованием логических выражений.

Что нужно знать:

- таблицы истинности логических операций «И», «ИЛИ», «НЕ» если в выражении нет скобок, сначала выполняются все операции «НЕ», затем – «И», затем – «ИЛИ»;
- логическое произведение $A \cdot B \cdot C \dots$ равно 1 (выражение истинно) только тогда, когда все сомножители равны 1 (в остальных случаях равно 0);
- логическая сумма $A + B + C + \dots$ равна 0 (выражение ложно) только тогда, когда все слагаемые равны 0 (а в остальных случаях равна 1);
- правила преобразования логических выражений:

Законы алгебры логики

21

название	для И	для ИЛИ
двойного отрицания	$\overline{\overline{A}} = A$	
исключения третьего	$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$
операции с константами	$A \cdot 0 = 0, A \cdot 1 = A$	$A + 0 = A, A + 1 = 1$
повторения	$A \cdot A = A$	$A + A = A$
поглощения	$A \cdot (A + B) = A$	$A + A \cdot B = A$
переместительный	$A \cdot B = B \cdot A$	$A + B = B + A$
сочетательный	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$
распределительный	$A + B \cdot C = (A + B) \cdot (A + C)$	$A \cdot (B + C) = A \cdot B + A \cdot C$
законы де Моргана	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$

- ввод какого-то слова (скажем, **тигры**) в запросе поисковой системы означает, что пользователь ищет Web-страницы, на которых встречается это слово;
- операция «И» всегда **ограничивает** поиск, то есть, в ответ на запрос **тигры И косули** поисковый сервер выдаст **меньше** страниц, чем на запрос **тигры**, потому что будет искать страницы, на которых есть оба этих слова одновременно;
- операция «ИЛИ» всегда **расширяет** поиск, то есть, в ответ на запрос **тигры ИЛИ косули** поисковый сервер выдаст **больше** страниц, чем на запрос, **тигры** потому что будет искать страницы, на которых есть хотя бы одно из этих слов (или оба одновременно);

- если в запросе вводится фраза в кавычках, поисковый сервер ищет страницы, на которых есть в точности эта фраза, а не просто отдельные слова; взятие словосочетания в кавычки **ограничивает** поиск, то есть, в ответ на запрос **"тигры косули"** поисковый сервер выдаст **меньше** страниц, чем на запрос **тигры косули**, потому что будет искать только те страницы, на которых эти слова стоят одно за другим.

ПРИМЕРЫ ЭКЗАМЕНАЦИОННЫХ ЗАДАНИЙ

А1. Четыре человека – Иванов, Петров, Андреев, Сидоров предполагают разыграть лотерею. В лотерейном барабане лежат 4 билета.

Иванов хочет тянуть билет первым или четвертым, Андреев – третьим или четвертым, Сидоров – первым или вторым, а Петров не хочет быть ни первым, ни последним.

Какой из предложенных вариантов очереди, удовлетворяет всем пожеланиям.

(Обозначения: И – Иванов, П – Петров, А – Андреев, С – Сидоров)

1) И П А С 2) С А П И 3) И С П А 4) С П И А

А2. В городе N в каждом школьном классе учится не более 30 человек.

Сколько двоичных цифр необходимо, чтобы записать в двоичной системе число учеников в каждом классе?

1) 5 2) 6 3) 7 4) 4

А3. Сколько существует различных последовательностей из символов «плюс» и «минус», длиной ровно в пять символов?

1) 64 2) 50 3) 32 4) 20

А4. Какой объем памяти в мегабайтах занимает фотография размером 1024x1024 точек, каждая из которой кодируется 24 битами?

- 1) 1
- 2) 24
- 3) 3
- 4) 3072

А5. Количество нулей в двоичной записи десятичного числа, которое можно представить в виде $4+16+64+256+1024$, равно

- 1) 5
- 2) 6
- 3) 3
- 4) 11

А6. В восьмеричной системе счисления после числа 777 следует число

- 1) 778
- 2) 780
- 3) 1000
- 4) 1777

А7. Верны ли следующие суждения о вспомогательном алгоритме?

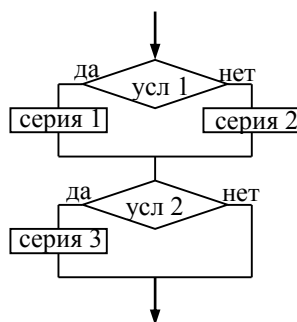
А. Если исполнителем является компьютер, то алгоритм записывается с помощью языка программирования.

Б. Алгоритм нельзя представить с помощью схемы, программы, алгоритмического языка, пронумерованных строк.

- 1) верно только А

- 2) верно только Б
- 3) оба суждения верны
- 4) оба суждения неверны

А8. Фрагмент блок-схемы



представляет алгоритм, который содержит

- 1) две команды ветвления, соединенных между собой последовательно
- 2) две команды ветвления, одна из которых вложена в другую
- 3) две команды ветвления сокращенной формы
- 4) две команды ветвления полной формы

А9. Определите значение целочисленных переменных a, b и c после выполнения фрагмента программы:

Бейсик	Паскаль	Алгоритмический
a=50 b=17 c=b*a\b a=a\b*b b=c-a ' \ - стандартная операция, вычисляющая результат деления нацело первого аргумента на второй	a:=50; b:=17; c:=b*a div b; a:=a div b*b; b:=c-a; {div - стандартная операция, вычисляющая результат деления нацело первого аргумента на второй}	a:=50 b:=17 c:=div(a*b, b) a:=div(a, b)*b b:=c-a div - стандартная функция, вычисляющая результат деления нацело первого аргумента на второй

- 1) a=34,b=16,c=50 2) a=50,b=0,c=50 3) a=34,b=0,c=50 4) a=50,b=-16,c=34

А10. Элементы двумерного массива А первоначально были обнулены. Затем значения элементов меняются с помощью вложенного оператора цикла в представленном фрагменте программы.

Бейсик	Паскаль	Алгоритмический
FOR n=1 TO 5 FOR k=1 TO n A(n,k)=A(n,k)+1 A(k,n)=A(k,n)+1	for n:=1 to 5 do for k:=1 to n do begin A[n, k]:=A[n, k]+1; end	нц для n от 1 до 5 нц для k от 1 до n A[n, k]:=A[n, k]+1 A[k, n]:=A[k, n]+1

NEXT k	A[k, n]:=A[k, n]+1;	кц
NEXT n	end	кц

Сколько элементов массива в результате будут равны 2?

- 1) 0 2) 5 3) 20 4) 25

A11. Для какого имени истинно высказывание

(Первая буква имени согласная \rightarrow Вторая буква имени согласная) \wedge Четвертая буква имени согласная

- 1) ФЕДОР 2) СЕРГЕЙ 3) АНТОН 4) ЕЛЕНА

A12. Какое логическое выражение равносильно $\neg(\neg A \vee B)$?

- 1) $A \wedge \neg B$ 2) $A \vee \neg B$ 3) $\neg A \wedge \neg B$ 4) $A \vee B$

A13. Символом F обозначено одно из указанных ниже логических выражений от трех аргументов: X, Y, Z.

Дан фрагмент таблицы истинности выражения F:

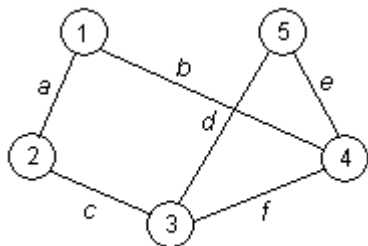
X	Y	Z	F
1	0	0	1
0	1	0	0
0	0	1	1

Чему равно F?

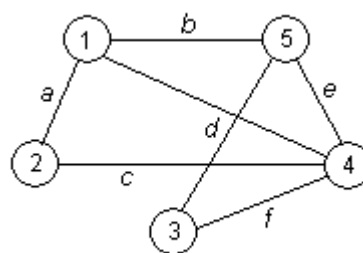
- 1) $X \vee Y \vee Z$ 2) $X \wedge Y \wedge Z$ 3) $X \wedge Y \wedge \neg Z$ 4) $X \vee \neg Y \vee Z$

A14. Укажите, какой из представленных на рисунке графов соответствует символической записи в виде списка ребер с указанием вершин, соединенных этими ребрами a(1,2) b(1,4) c(2,4) d(3,5) e(4,5) f(3,4).

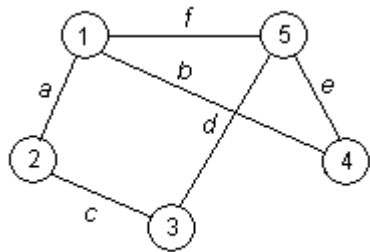
1)



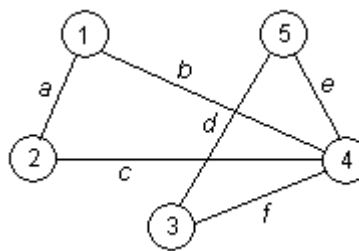
2)



3)



4)



A15. Символу А в двоичной системе счисления соответствует код 0100 0001, символу В – 0100 0010, символу “+” – 0011 1011. Найти код последовательности символов А+В

- 1) 01000001 00111011 01000010
- 2) 01000001 01111011 01000010
- 3) 01000001 01000010 00111011
- 4) 00111011 01000010 01000001

A16. Цепочка из трех бусин формируется по следующему правилу. В середине цепочки стоит одна из бусин А, В, Г. На третьем месте – одна из бусин Б, В, Г. На первом месте – одна из бусин А, Б, В, которой нет на втором или третьем месте.

Какая из следующих цепочек создана по этому правилу?

- 1) БГА
- 2) ВАВ
- 3) АВВ
- 4) ВБГ

В1. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 21 оканчивается на 0.

В2. Укажите значения переменных К, L, M, N при которых логическое выражение $(K \rightarrow M) \wedge (\neg L \vee \neg M) \wedge \neg N \wedge K$ истинно

Ответ запишите в виде строки из четырех символов: значений переменных К, L, M и N (в указанном порядке). Так, например, строка 0101 соответствует тому, что $K=0, L=1, M=0, N=1$.

В3. У исполнителя Удвоитель две команды, которым присвоены номера:

1. вычти 1,
2. умножь на 2.

Первая из них уменьшает число на экране на 1, вторая – удваивает его.

Запишите порядок команд в программе получения из 3 числа 16, содержащей не более 4 команд, указывая лишь номера команд. (Например, программа 21211 – это программа:

- умножь на 2,
- вычти 1,
- умножь на 2,
- вычти 1,

вычти 1, которая преобразует число 1 в 0).

В4. Ваш компьютер подключен к Интернет через локальную сеть. При загрузке в браузер некоторой web-страницы вы видите, что она содержит устаревшую информацию. Что следует предпринять для получения актуальной информации?

- 1) исправить системные дату и время компьютера
- 2) выполнить команду браузера “обновить страницу”
- 3) отсоединить компьютер от локальной сети и подсоединить снова
- 4) закрыть браузер, затем открыть его заново и снова загрузить страницу

С1. На координатной плоскости задана прямая $y=k_1x$, причем известно, что k_1 - целое число, большее нуля. С клавиатуры вводится значение k_1 , а также целые числа x_0, y_0 , задающие положение точки $P(x_0, y_0)$, причем известно, что точка P лежит внутри 1-го квадранта (т.е. $x_0 > 0, y_0 > 0$). В программе требовалось проверить, лежит ли P выше заданной прямой, на прямой, или ниже прямой, с выдачей на экран соответствующих сообщений.

Была написана программа, приведенная ниже. В программе используется переменная k_0 типа `real`.

1) Внести изменения в программу определения положения точки P относительно прямой $y=k_1x$, чтобы эта программа использовала только переменные типа `integer`.

2) Постановка задачи изменилась: теперь, кроме первой прямой, имеется вторая прямая $y=k_2x$, (k_2 - также целое число, большее нуля). Требуется доработать программу так, чтобы она определяла, лежит ли точка P строго между этими двумя прямыми ("строго между" означает, что точка не должна лежать ни на одной из этих прямых). На экран выдать выходное сообщение (лежит между или не лежит между). Переменные типа `real` можно использовать.

3) Доработать программу так же, как указано в п.2, но программа может использовать только переменные типа `integer`.

Примечание: баллы выставляются независимо за каждый из правильно выполненных 3-х пунктов задания.

Программа на Паскале	Программа на Бейсике
<pre>VAR x0,y0: integer; k1,k2: integer; k0: real; i: integer; BEGIN readln(x0,y0,k1); k0:=y0/x0; if k0>k1 then writeln ('P выше прямой'); if k0=k1 then writeln ('P на прямой'); if k0<k1 then writeln ('P ниже прямой'); END.</pre>	<pre>CLS INPUT x0, y0, k1 k0=y0/x0 IF k0>k1 THEN PRINT "P выше прямой" IF k0=k1 THEN PRINT "P на прямой" IF k0<k1 THEN PRINT "P ниже прямой" END</pre>

C2. Имеется программа, которая сначала заполняет массив A длиной $N=400$ подряд идущими натуральными числами от 1 до 400, а затем выполняет некоторые действия над элементами массива, используя цикл.

(Текст программы приведен ниже).

- 1) Какое значение будет иметь элемент $A[1]$ по окончании программы?
- 2) Пусть мы в программе изменили длину массива, и теперь $N=401$. Какое значение будет в этом случае иметь элемент $A[1]$ по окончании программы?
- 3) Пусть в программе N (длина массива) - такое натуральное число, что выполнено равенство $N \bmod 4 = k$ (то есть N при делении на 4 дает остаток k). Требуется указать все такие значения k , при которых по окончании программы элемент $A[1]$ имеет нулевое значение.

Примечание: баллы выставляются независимо за каждый из правильно выполненных 3-х пунктов задания.

Программа на Паскале	Программа на Бейсике
<pre> CONST N=400; { длина массива } VAR A: array[1..N] of integer; i: integer; BEGIN for i:=1 to N do A[i]:=i; for i:=N downto 2 do begin A[i-1]:=abs(A[i-1]-A[i]); A[i]:=0; end; END.</pre>	<pre> CLS N=400 'длина массива DIM A(N) FOR i=1 TO N A(i)=i NEXT i FOR i=N TO 2 STEP -1 A(i-1)=ABS(A(i-1)-A(i)) A(i)=0 NEXT i END</pre>

СПИСОК РЕКОМЕНДОВАННОЙ ЛИТЕРАТУРЫ

1. Глизбург В., Самойлова Е. Информатика и ИКТ. Комплексная подготовка. М.: Айрис Пресс, 2013 г. – 336 с.
2. Каймин В.А. Информатика. Учебник для поступающих. М.: Проспект, 2012 – 171с.
3. Каймин В.А., Касаев Б.С. Информатика: Практикум на ЭВМ., М, ИНФРА-М, 2009 – 201 с.
4. Чуркина Т.Е. Информатика. Подготовка к ЕГЭ. Экзамен, 2011 г. – 256 с.
5. Макарова Н.В. Информатика. Практикум по работе на компьютере. М., Финансы и Статистика, 2011 – 206 с.
6. Вовк Е.Т. Информатика. Пособие для подготовки к ЕГЭ // Вовк Е.Т., Глинка Н.В., Грацианова Т.Ю. и др. 2-е изд., испр. и доп. – М.: 2013. - 322 с.
7. ЕГЭ 2016. Официальный информационный портал единого государственного экзамена. // Url: <http://www.ege.edu.ru/>
8. ЕГЭ по информатике и ИКТ // <http://5ballov.qip.ru/test/ege/informatika-i-ikt>