**Билет 1**

1. Log2(30) = 4,91 = 5  
   Log2(7) = 2,81 = 3  
   5+3 = 8
2. Log2(30) = 4,91 = 5  
   Log2(7) = 2,81 = 3  
   5+3 = 8
3. По форме представления: символьная (дорожный знак), текстовая (книга), графическая (фото), звуковая (аудиофайлы), видео (фильмы).

По способу восприятия: визуальная (графики), аудиальная (музыка), тактильная (текстура ткани), обонятельная (запах цветов), вкусовая (еда).

По общественному значению: массовая (новости), специальная (научные статьи), личная (паспорт).

1. 168,375 из 10чной в 2чную в 16чную

0,1010 1000 011

1000000 + 1000 = 1001000

0100 1000 1010 1000 0110 0000 0000 0000

4 8 A 8 6 0 0 016

1. import math

def task1(n, m,):

k = 1

V = n \* m \*

return int (V)

if name == "main" :

n = int (input("Ширина: "))

m = int (input("Высота: "))

print (task1(n, m)

1. Физ. природа: механические, акустические, оптические, электрические, проводные, беспроводные.

По направлению передачи: симплексный (передача в одном направлении), дуплексный (передача в двух направлениях одновременно), полудуплексный (попеременная передача).

По диапазону частот: низкочастотный, среднечастотный, высокочастотный, сверхвысокочастотный.

По типу сигнала: аналоговый (непрерывный сигнал, который изменяется во времени), цифровой (дискретный сигнал, представляющий последовательность двоичных значений)

Время существования: коммутируемые (создаются только на время передачи, после уничтожаются), некоммутируемые (создаются на длительное время)

Скорость передачи инф: низкоскоростные, среднескоростные, высокоскоростные.

**Билет 2**

1. Log2(67) = 6,07 бит
2. 1024 / 8 = 128 кб = 128 \* 1024 = 131072 байт = 131072 \* 8 = 1048576 бита

1048576 / 4096 = 256 бита на символ

4096 / 256 = 16 символов

1. Процесс дискретизации информации — это процесс перевода непрерывного аналогового сигнала в цифровой формат. Примеры использования: запись музыки, цифровая фотография, сотовая связь.
2. D – 1101 9 1001 7 0111 1 0001 1 0001

1101 1001 0111 0001 0001

1011001 – 1000000 = 11001

0111 000,1 0001 = - 56,53125

1. def task3(nn, mm, vv):

vv = vv \* 1024 \* 8

k = vv / (nn \* mm)

return int (k)

nn = int (input("Ширина: "))

mm = int (input("Высота: "))

vv = int (input("Объем памати: "))

print (task3(nn, mm, vv))

1. Непрерывные - предназначенные для передачи непрерывных сигналов.

Дискретные — предназначенные для передачи дискретных сигналов.

Симплексные — данные передаются только в одном направлении.

Дуплексные — данные могут передаваться в один и тот же момент времени в двух направлениях.

Полудуплексные — данные могут передаваться поочерёдно в прямом и обратном направлении.

**Билет 3**

1. I = k \*\* i

N = 31 \* 12 \* 24 = 8928

I = log2(8928) = 13,12 бит

1. 8775 байт = 8775 \* 8 бит = 70200 байт

N = 2i

30 = 2i

i = 6

n = i / 6 \* 6 \* 30 = 70200 / 1080 = 65 символов в строке

1. В природе: Звуковые волны от песни птицы, канал передачи – воздух, приемник - птицы или животные

Телефонная связь: Человеческий голос, передатчик – микрофон, канал передачи - телефонная сеть, приемник - динамик на телефоне, получатель – человек.

1. 254,1285

0.1111 1110 0010 0000 1110

1000000 + 1000 = 1001000

0100 1000 1111 1110 0010 0000 1110 0000

4 8 F E 2 0 E 016

1. def task4 (k, vvv):

vvv = vvv \* 8

k = math.log2 (k)

mn = vvv / k

return int (mn)

k = int (input("кол-во цветов: "))

vvv = int (input("объем памяти: "))

print (task4(k, vvv))

1. Аналоговые каналы: передача инф в непрерывной форме.

Цифровые каналы: передача инф в дискретной форме, в виде последовательности битов

**Билет 4**

1. N=2H

N=264

N=1.8\*1019

1. 8775 байт \* 8 бит/байт = 70200 бит 2^6 = 64

70200 бит / 6 бит/символ = 11700 символов

6 страниц \* 30 строк/страницу = 180 строк

11700 символов / 180 строк = 65 символов

1. По форме представления: символьная (представляет информацию в виде чисел), текстовая (используется для представления букв, цифр, знаков препинания, других символов), графическая (представляет информацию в виде изображений, схем, чертежей, диаграмм, графиков), звуковая (использует звуковые волны для представления речи, музыки, звуков), видео (представляет движущееся изображение со звуком).
2. E – 1110 5 0101 6 0110 1 0001 2 0010

1110 0101 0110 0001 0010

1100101 – 1000000 = 100101

0110 00,01 0010

24,28125

1. def task3(v2, resolution2, rate2):

v2 = v2 \* (1024 \*\* 3)

time2 = v2 / (rate2 \* resolution2 \* (8))

return float(time2)

v2 = float(input("объем в гигибайтов (через точку): "))

rate2 = int(input("частота: "))

resolution2 = int(input("разрешение: "))

print (task3(v2, rate2, resolution2))

перевод гб в байт, делим байты на частоту умноженную на разрядность на 8

1. Коммутируемые (временные), которые создаются только на срок передачи информации.

Некоммутируемые (выделенные), которые создаются на длительный период и имеют постоянные характеристики.

**Билет 5**

1. 26=64 этажа
2. 2\*1,5\*1023= 3\*10\*23
3. Количество информации — это мера неопределенности, которая уменьшается при получении новых данных. Чем больше неопределенности, тем больше информации мы получаем, когда ее устраняем.
4. 113,475

0.1110 0010 1111 0011 0010

1000000 + 0111 = 1000111

0100 0111 1110 0010 1111 0011 0010 0000

4 7 E 2 F 3 2 016

1. def task3(v2, resolution2, rate2):

v2 = v2 \* 1024 #Перевод килобайты в байты

time2 = v2 / (rate2 \* resolution2 \* (8)) #Находим время звучания в секундах

return float(time2)

v2 = float(input("объем в гигибайтов (писать через точку): "))

rate2 = int(input("частота: "))

resolution2 = int(input("разрешение: "))

print (task3(v2, rate2, resolution2))

перевод клб в байт, делим байты на частоту умноженную на разрядность на 8

1. Модуляция — это процесс изменения одного или нескольких параметров модулируемого несущего сигнала при помощи модулирующего сигнала

**Билет 6**

1. Log2(32)= 5 бит
2. I = k \* i

I = 2 \* k

I = 7 \* k (128 символов)

7 / 2 = 3.5

помехи

приемник

источник

Канал связи

Декод устр

Кодирующее устр

Защита от помех

Информация кодируется с помощью кодировщика в форму, подходящую для передачи по каналу связи. Канал связи может исказить информацию из-за шумов и помех – защита от помех. Декодировщик восстанавливает исходную информацию. Приемник получает информацию.

1. F9811 по двоично-шестнадтцатиричной системе переводим каждое число отдельно и дополняем нулями

F- 1111 9 1001 8 1000 1 0001 1 0001

1 111 1001 1000 0001 0001

1111 001-1000000 = 11101

1000 000,1 0001 = - 64,5312510

1. 3 \* 60 = 180 секунд  
   16 \* 1000 = 16000  
   180 \* 16000 \* 32 / 8 = 11520000 байт   
   11520000 / 1024 = 11250 килобайт
2. Протокол передачи данных - это набор правил, описывающих форматы, процедуры и последовательность действий, необходимых для передачи информации между двумя или более устройствами.

**Билет 7**

1. Log2(25) = 5   
   Log2(8) = 3  
   5+3 = 8
2. 3кб = 3072 байтов

3072 / 3072 = 1 символа

1. Информация — это любые сведения, которые уменьшают неопределенность, расширяют знания или влияют на принятие решений.

Данные — это сырые факты и значения, не имеющие самостоятельного смысла до тех пор, пока они не будут обработаны и интерпретированы.

Знания — это интерпретированная и структурированная информация, которая позволяет понимать мир, принимать решения и решать задачи.

Данные преобразуются в информацию с помощью обработки и интерпретации. Из информации получаются знания путем структурирования, связывания с другими данными и осмысления.

Пример:

Данные: Температура воздуха 25 градусов Цельсия.

Информация: В комнате тепло.

1. 178,325

0.1011 0010 0101 0011 0011

1000000 + 1000 = 1001000

0100 1000 1011 0010 0101 0011 0011 0000

4 8 B 2 5 3 3 016

1. def task4 (k, vvv):

vvv = vvv \* 8 #перевод из байтов в бит

k = math.log2 (k) #узнаём кол-во бит в одной точке

mn = vvv / k #кол-во точек на экране

return int (mn)

k = int (input("кол-во цветов: "))

vvv = int (input("объем памяти: "))

print (task4(k, vvv))

1. Основание системы счисления — это количество цифр или других символов, необходимых для написания числа в этой системе счисления.

**Билет 8**

1. Log2(63) = 5.98 бит
2. 0.25 мб = 2097152 бит

2097152 / 2018 = 1039,223 символа

1. Непрерывность – возможность накапливать информацию

Дискретность – инф состоит из отдельных частей

1. D9721

D 1101 9 1001 7 0111 2 0010 1 0001

1101 1001 0111 0010 0001

1011001-1000000 = 11001

0111 001.0 0001 = -57.03125

1. def task3(nn, mm, vv):

vv = vv \* 1024 \* 8 #Перевод Килобайтов в биты

k = vv / (nn \* mm) #определение кол-во цветов

return int (k)

nn = int (input("Ширина: "))

mm = int (input("Высота: "))

vv = int (input("Объем памати: "))

print (task3(nn, mm, vv))

1. Разделение на целую и дробную части.

Перевод целой части. Перевод дробной части. Соединение частей.

**Билет 9**

1. N=30\*12\*24=8640

I = log2(8640) = 13,07

1. 6348 байт \* 8 бит/байт = 50784 бит

26=64 6\*30=180 строк 8464 /180 = 47 символов

1. Формат с фиксированной точкой. В нём представляются только целые числа. Формат с плавающей точкой. В нём представлены вещественные числа (дробные).

Целые числа в памяти компьютера хранятся в формате с фиксированной запятой.

1. 254,1285 из 10 сс в 2 сс в 16 сс

0.1111 1110 0010 0000 1110

1000000 + 1000 = 1001000

0100 1000 1111 1110 0010 0000 1110 0000

4 8 F E 2 0 E 016

1. 27=128 7=1 байт 360\*1=360
2. Зная вероятность вхождения символов в сообщение, можно описать процедуру построения кодов переменной длины, состоящих из целого количества битов.

**Билет 10**

1. N=2H

N=2128

N=3,4\*1038

1. 8\*256= 2048

256 байт 8бит/байт = 2048 бит

2048бит= 2048 символ

Log2(N) = 2048/2048

Log2(N) = 21 = 2 мощ

1. Заключается в оптимальном кодировании символов с помощью двоичных кодов, где более частые символы получают более короткие коды, а менее частые - более длинные.
2. E5632

E- 1110 5- 0101 6-0110 3- 0011 2- 0010

1110 0101, 0110 0011 0010

1100101-1000000 = 100101

0110 00,11 0010 = -24,7812510

1. def task3(v2, resolution2, rate2):

v2 = v2 \* (1024 \*\* 3) #Перевод гигабайтов в байты

time2 = v2 / (rate2 \* resolution2 \* (8)) #Находим время звучания в секундах

return float(time2)

v2 = float(input("объем в гигибайтов (писать через точку): "))

rate2 = int(input("частота: "))

resolution2 = int(input("разрешение: "))

print (task3(v2, rate2, resolution2))

1. Изображение выглядит как текст, диаграмма, снимок экрана, Шрифт

   Автоматически созданное описание

**Билет 11**

1. 28= 256 этажей
2. 2\*1,5\*1023= 3\*10\*23
3. Бит, байт, килобайт, мегабайт, гигабайт, терабайт.

1 Кбайт = 1024 байт = 8192 бит

1 Мбайт = 1024 Кбайт = 1048576 байт = 8388608 бит

1 Гбайт = 1024 Мбайт = 1073741824 байт = 8589934592 бит

1 Тбайт = 1024 Гбайт = 1099511627776 байт = 8796093022208 бит

1. 123,425 из 10 сс в 2 сс в 16 сс

0,1111 0110 0110 1100 1100

1 1111 0110 0110 1100 1100 0000

1. F 6 6 C C 016
2. def task3(v2, resolution2, rate2):

v2 = v2 \* 1024 #Перевод килобайты в байты

time2 = v2 / (rate2 \* resolution2 \* (8)) #Находим время звучания в секундах

return float(time2)

v2 = float(input("объем в гигибайтов (писать через точку): "))

rate2 = int(input("частота: "))

resolution2 = int(input("разрешение: "))

print (task3(v2, rate2, resolution2))

1. Актуальность - информация актуальна, если она относится к данному моменту времени и имеет значение для принятия решений.

Достоверность - достоверная информация является истинной и подтвержденной фактами.

Полнота - информация полноценна, если она содержит все необходимые сведения для понимания ситуации или принятия решения.

Релевантность - релевантная информация имеет прямое отношение к конкретной задаче или вопросу.

Ценность - ценность информации определяется ее полезностью для конкретного получателя и возможностью использовать ее для достижения целей.

Понятность - информация понятна, если она представлена в доступной форме и использует понятные для получателя термины и концепции.

Своевременность - информация своевременна, если она предоставляется в нужный момент времени и не устаревает до того, как ее используют.

**Билет 12**

1. Log2(32)= 5 бит
2. I = k \* i

I = 2 \* k

I = 7 \* k (128 символов)

7 / 2 = 3.5

1. Символьная – используется для представления в виде букв, цифр, знаков.

Текстовая – используется для представления сочетания букв, цифр, знаков препинания и тд.

Графическая – представляет инфу в виде изображений, схем, чертежей, диаграмм, графиков.

Звуковая – использует звуковые волны для представления речи, музыки, звуков.

Видео - представляет движущееся изображение со звуком.

1. F9811 1111 9 1001 8 1000 1 0001 1 0001

1 111 1001 1000 0001 0001

1111 001-1000000 = 11101

1000 000,1 0001 = - 64,5312510

1. 3 \* 60 = 180 секунд  
   16 \* 1000 = 16000  
   180 \* 16000 \* 32 / 8 = 11520000 байт   
   11520000 / 1024 = 11250 килобайт
2. Разделить двоичное число на целую и дробную части. Перевести целую часть (Восьмеричную: сгруппировать по 3 бита, начиная с младшего разряда (дополняя нулями при необходимости). Шестнадцатеричную: сгруппировать по 4 бита, начиная с младшего разряда (дополняя нулями при необходимости)). Перевести дробную часть (Восьмеричную: добавьте нули справа, чтобы количество битов было кратно 3. Сгруппируйте по 3 бита, начиная с целой части. Переведите каждую группу. Шестнадцатеричную: добавьте нули справа, чтобы количество битов было кратно 4. ) Сгруппируйте по 4 бита, начиная с целой части. Переведите каждую группу. Пример:

10110,1101₂ = 52,64₈ = 16,D0₁₆

**Билет 13**

1. Log2(30) = 4,91 = 5  
   Log2(7) = 2,81 = 3  
   5+3 = 8
2. 1.5кб = 1536 байтов

3072 / 1536 = 2 символа

1. По форме представления: символьная (буквы, цифры, знаки), текстовая (сочетания букв, цифр, матем знаков), графическая (схемы, изображения), звуковая (музыка, аудиофайлы), видео (изображение со звуком)

По способу восприятия: визуальная (графики), аудиальная (музыка), тактильная (текстура ткани), вкусовая (еда), обонятельная (запах).

По массовому значению: массовая (новости), специальная (научные статьи), личная (паспорт).

1. 168,375 из 10 сс в 2 сс в 16 сс

0,1010 1000 011

1000000 + 1000 = 1001000

0100 1000 1010 1000 0110 0000 0000 0000

4 8 A 8 6 0 0 016

1. import math

def task1(n, m,):

k = 1

V = n \* m \* k #Находим объем памяти данного изображения

return int (V)

if name == "main" :

n = int (input("Ширина: "))

m = int (input("Высота: "))

print (task1(n, m)

1. Физ. природа: механические, акустические, оптические, электрические, проводные, беспроводные.

По направлению передачи: симплексный (передача в одном направлении), дуплексный (передача в двух направлениях одновременно), полудуплексный (попеременная передача).

По диапазону частот: низкочастотный, среднечастотный, высокочастотный, сверхвысокочастотный.

По типу сигнала: аналоговый (непрерывный сигнал, который изменяется во времени), цифровой (дискретный сигнал, представляющий последовательность двоичных значений)

Время существования: коммутируемые (создаются только на время передачи, после уничтожаются), некоммутируемые (создаются на длительное время)

Скорость передачи инф: низкоскоростные, среднескоростные, высокоскоростные.

**Билет 14**

1. Log2(67) = 6,07 бит
2. 1/512 Мбайта = 2048 байт

2048\*8=16384

16384/4096=4 24=16

1. Дискретизация — это преобразование непрерывного сигнала (аналогового) в дискретный (цифровой), путем отбора значений сигнала в определенные моменты времени и их записи в виде цифровых значений.

Примеры: запись музыки на CD, воспроизведение звука, создание цифровых фотографий, передача голоса, видео, данных по сети.

1. D9721

D 1101 9 1001 7 0111 2 0010 1 0001

1101 1001 0111 0010 0001

1011001-1000000 = 11001

0111 001.0 0001 = -57.03125

1. def task3(nn, mm, vv):

vv = vv \* 1024 \* 8 #Перевод Килобайтов в биты

k = vv / (nn \* mm) #определение кол-во цветов

return int (k)

nn = int (input("Ширина: "))

mm = int (input("Высота: "))

vv = int (input("Объем памати: "))

print (task3(nn, mm, vv))

1. Непрерывные - предназначенные для передачи непрерывных сигналов.

Дискретные — предназначенные для передачи дискретных сигналов.

Симплексные — данные передаются только в одном направлении.

Дуплексные — данные могут передаваться в один и тот же момент времени в двух направлениях.

Полудуплексные — данные могут передаваться поочерёдно в прямом и обратном направлении.

**Билет 15**

1. I = k \*\* i

N = 31 \* 12 \* 24 = 8928

I = log2(8928) = 13,12 бит

1. 8775 байт \* 8 бит/байт = 70200 бит

70200 бит / 6 бит/символ = 11700

26=64 6\*30=180

11700 / 180 = 65 символов/строку

1. Передача информации в природе:

Источник: Солнце, излучающее тепло и свет.

Кодирование: Электромагнитное излучение с различной длиной волны.

Канал связи: Космическое пространство.

Декодирование: Глаза животных и растений, чувствительные к тепловому и световому излучению.

Приемник: Живые организмы.

Передача информации в технике:

Источник: Компьютер, генерирующий цифровой сигнал.

Кодирование: Модуляция цифрового сигнала на несущую частоту (например, Wi-Fi).

Канал связи: Радиоволны, оптический кабель, электрический провод.

Декодирование: Демодуляция сигнала на приёмнике.

Приемник: Другой компьютер, мобильный телефон и т.д., воспринимающий и обрабатывающий полученную информацию.

1. 254,1285 из 10 сс в 2 сс в 16 сс

0.1111 1110 0010 0000 1110

1000000 + 1000 = 1001000

0100 1000 1111 1110 0010 0000 1110 0000

4 8 F E 2 0 E 016

1. def task4 (k, vvv):

vvv = vvv \* 8 #перевод из байтов в бит

k = math.log2 (k) #узнаём кол-во бит в одной точке

mn = vvv / k #кол-во точек на экране

return int (mn)

k = int (input("кол-во цветов: "))

vvv = int (input("объем памяти: "))

print (task4(k, vvv))

1. Аналоговые каналы: передача инф в непрерывной форме.

Цифровые каналы: передача инф в дискретной форме, в виде последовательности битов.

**Билет 16**

1. N=2H

N=264

N=1.8\*1019

1. 8775 байт \* 8 бит/байт = 70200 бит 2^6 = 64

70200 бит / 6 бит/символ = 11700 символов

6 страниц \* 30 строк/страницу = 180 строк

11700 символов / 180 строк = 65 символов

1. Символьная – используется для представления в виде букв, цифр, знаков.

Текстовая – используется для представления сочетания букв, цифр, знаков препинания и тд.

Графическая – представляет инфу в виде изображений, схем, чертежей, диаграмм, графиков.

Звуковая – использует звуковые волны для представления речи, музыки, звуков.

Видео - представляет движущееся изображение со звуком.

1. E5612 – 1110 5 – 0101 6 – 0110 1 – 0001 2 – 0010

1110 0101 0110 0001 0010

1100101 – 1000000 = 100101

0110 00,01 0010

24,28125

1. def task3(v2, resolution2, rate2):

v2 = v2 \* (1024 \*\* 3) #Перевод гигабайтов в байты

time2 = v2 / (rate2 \* resolution2 \* (8)) #Находим время звучания в секундах

return float(time2)v2 = float(input("объем в гигибайтов (писать через точку): "))

rate2 = int(input("частота: "))

resolution2 = int(input("разрешение: "))

print (task3(v2, rate2, resolution2))

1. Постоянные -каналы связи, которые существуют постоянно и не прерываются.

Временные - каналы связи, которые используются только в определенное время.

Условно-постоянные - каналы связи, которые могут быть прерваны на короткое время, например, из-за технических неисправностей или погодных условий.

**Билет 17**

1. 26=64 этажа
2. 2\*1,5\*1023= 3\*10\*23
3. Количество информации — это мера неопределенности, которая уменьшается при получении новых данных. Чем больше неопределенности, тем больше информации мы получаем, когда ее устраняем.
4. 113,475 из 10 сс в 2 сс в 16 сс

0.1110 0010 1111 0011 0010

1000000 + 0111 = 1000111

0100 0111 1110 0010 1111 0011 0010 0000

4 7 E 2 F 3 2 016

1. def task3(v2, resolution2, rate2):

v2 = v2 \* 1024 #Перевод килобайты в байты

time2 = v2 / (rate2 \* resolution2 \* (8)) #Находим время звучания в секундах

return float(time2)

v2 = float(input("объем в гигибайтов (писать через точку): "))

rate2 = int(input("частота: "))

resolution2 = int(input("разрешение: "))

print (task3(v2, rate2, resolution2))

1. Модуляция — это процесс изменения одного или нескольких параметров модулируемого несущего сигнала при помощи модулирующего сигнала.

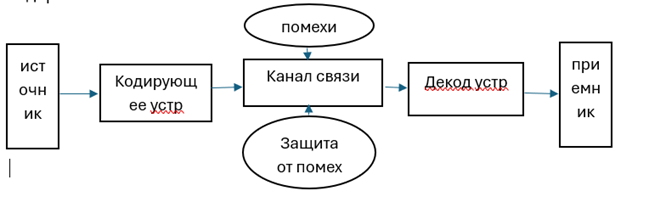
**Билет 18**

1. Log2(32)= 5 бит
2. I = k \* i

I = 2 \* k

I = 7 \* k (128 символов)

7 / 2 = 3.5

1. 

Информация кодируется с помощью кодировщика в форму, подходящую для передачи по каналу связи. Канал связи может исказить информацию из-за шумов и помех – защита от помех. Декодировщик восстанавливает исходную информацию. Приемник получает информацию.

1. F9811 F9811- 1111 9- 1001 8-1000 1-0001 1-0001

1 111 1001 1000 0001 0001

1111 001-1000000 = 11101

1000 000,1 0001 = - 64,5312510

1. 3 \* 60 = 180 секунд  
   16 \* 1000 = 16000  
   180 \* 16000 \* 32 / 8 = 11520000 байт   
   11520000 / 1024 = 11250 килобайт
2. Протокол передачи данных — это набор правил, описывающих форматы, процедуры и последовательность действий, необходимых для передачи информации между двумя или более устройствами.

**Билет 19**

1. Log2(25) = 5   
   Log2(8) = 3  
   5+3 = 8
2. 3кб = 3072 байтов

3072 / 3072 = 1 символа

1. Информация — это любые сведения, которые уменьшают неопределенность, расширяют знания или влияют на принятие решений.

Данные — это сырые факты и значения, не имеющие самостоятельного смысла до тех пор, пока они не будут обработаны и интерпретированы.

Знания — это интерпретированная и структурированная информация, которая позволяет понимать мир, принимать решения и решать задачи.

Данные преобразуются в информацию с помощью обработки и интерпретации. Из информации получаются знания путем структурирования, связывания с другими данными и осмысления.

Пример:

Данные: Температура воздуха 25 градусов Цельсия.

Информация: В комнате тепло.

Знания: если температура в комнате более 25 градусов Цельсия, то можно открыть окно, чтобы прохладный воздух проник внутрь.

1. 178,325 из 10 сс в 2 сс в 16 сс

0.1011 0010 0101 0011 0011

1000000 + 1000 = 1001000

0100 1000 1011 0010 0101 0011 0011 0000

4 8 B 2 5 3 3 016

1. def task4 (k, vvv):

vvv = vvv \* 8 #перевод из байтов в бит

k = math.log2 (k) #узнаём кол-во бит в одной точке

mn = vvv / k #кол-во точек на экране

return int (mn)

k = int (input("кол-во цветов: "))

vvv = int (input("объем памяти: "))

print (task4(k, vvv))

1. Основание системы счисления — это количество цифр или других символов, необходимых для написания числа в этой системе счисления.

**Билет 20**

1. Log2(63) = 5.98 бит
2. 0.25 мб = 2097152 бит

2097152 / 2018 = 1039,223 символа

1. Непрерывность – возможность накапливать информацию

Дискретность – инф состоит из отдельных частей

1. D9721- 1101 9- 1001 7-0111 2-0010 1-0001

1101 1001 0111 0010 0001

1011001-1000000 = 11001

0111 001.0 0001 = -57.03125

1. def task3(nn, mm, vv):

vv = vv \* 1024 \* 8 #Перевод Килобайтов в биты

k = vv / (nn \* mm) #определение кол-во цветов

return int (k)

nn = int (input("Ширина: "))

mm = int (input("Высота: "))

vv = int (input("Объем памати: "))

print (task3(nn, mm, vv))

1. Разделение на целую и дробную части.

Перевод целой части. Перевод дробной части. Соединение частей.

**Билет 21**

1. N=30\*12\*24=8640

I = log2(8640) = 13,07

1. 6348 байт \* 8 бит/байт = 50784 бит

26=64 6\*30=180 строк 8464 /180 = 47 символов

1. Формат с фиксированной точкой. В нём представляются только целые числа. Формат с плавающей точкой. В нём представлены вещественные числа (дробные).

Целые числа в памяти компьютера хранятся в формате с фиксированной запятой.

1. 254,1285 из 10 сс в 2 сс в 16 сс

0.1111 1110 0010 0000 1110

1000000 + 1000 = 1001000

0100 1000 1111 1110 0010 0000 1110 0000

4 8 F E 2 0 E 016

1. 27=128 7=1 байт 360\*1=360
2. Основная идея алгоритма Хаффмана состоит в том, что, зная вероятность вхождения символов в сообщение, можно описать процедуру построения кодов переменной длины, состоящих из целого количества битов.

***Билет номер 22***

**1.**

N=2H

N=2128

N=3,4\*1038

**2.**

8\*256= 2048

256 байт 8бит/байт = 2048 бит

2048бит= 2048 символ

Log2(N) = 2048/2048

Log2(N) = 21 = 2 мощ

**3.**

Как можно эффективнее и быстрее закодировать сообщение.

**4.**

E5632 по двоично-шестнадтцатиричной системе переводим каждое число отдельно и дополняем нулями

E- 1110 5- 0101 6-0110 3- 0011 2- 0010

1110 0101, 0110 0011 0010

1100101-1000000 = 100101

0110 00,11 0010 = -24,7812510

**5.**

def task3(v2, resolution2, rate2):

    v2 = v2 \* (1024 \*\* 3)                #Перевод гигабайтов в байты

    time2 = v2 / (rate2 \* resolution2 \* (8))   #Находим время звучания в секундах

    return float(time2)

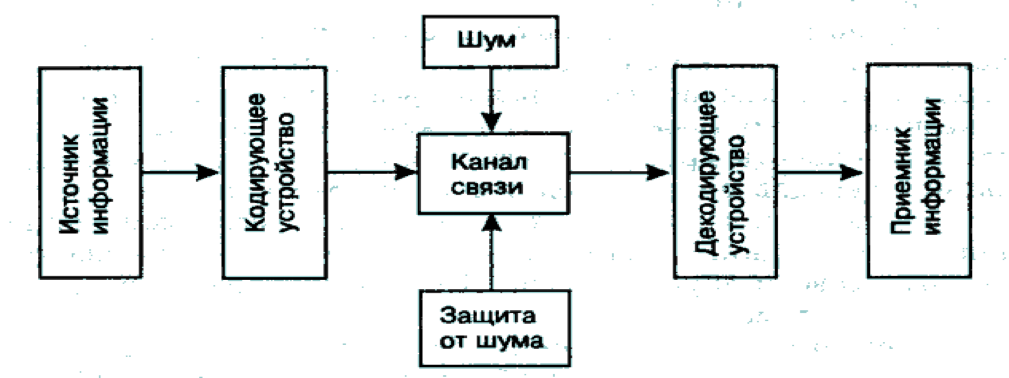
v2 = float(input("объем в гигибайтов (писать через точку): "))

    rate2 =  int(input("частота: "))

    resolution2 = int(input("разрешение: "))

    print (task3(v2, rate2, resolution2))

перевод гб в байт, делим байты на частоту умноженную на разрядность на 8

**6.**

***Билет номер 23***

**1.**

28= 256 этажей

**2.**

2\*1,5\*1023= 3\*10\*23

**3.**

Бит, байт, килобайт, мегабайт, гигабайт, терабайт

1/8 1/1024 1/1024 1/1024 1/1024 1/1024

**4.**

123,425 из 10 сс в 2 сс в 16 сс

0,1111 0110 0110 1100 1100

0100 0111   1111 0110   0110 1100   1100 0000

  4        7          F       6          6         C         C       016

**5.**

def task3(v2, resolution2, rate2):

    v2 = v2 \* 1024                 #Перевод килобайты в байты

    time2 = v2 / (rate2 \* resolution2 \* (8))  #Находим время звучания в секундах

    return float(time2)

v2 = float(input("объем в гигибайтов (писать через точку): "))

    rate2 =  int(input("частота: "))

    resolution2 = int(input("разрешение: "))

    print (task3(v2, rate2, resolution2))

перевод клб в байт, делим байты на частоту умноженную на разрядность на 8

**6.**

Актуальность, полнота, доступность, понятность, надежность, значимость, динамичность, полезность, экономичность

***Билет номер 24***

**1.**

Log2(32)= 5 бит

**2.**

I = k \* i

1) I = 2 \* k

2) I = 7 \* k (128 символов)

3) 7 / 2 = 3.5

**3.**

Звуковые, графические, видео, символьные, текстовые

**4.**   
F9811 по двоично-шестнадтцатиричной системе переводим каждое число отдельно и дополняем нулями

F- 1111 9- 1001 8-1000 1-0001  1-0001

1 111 1001 1000 0001 0001

1111 001-1000000 = 11101

1000 000,1 0001 = - 64,5312510

**5.**

3 \* 60 = 180 секунд  
16 \* 1000 = 16000  
180 \* 16000 \* 32 / 8 = 11520000 байт   
11520000 / 1024 = 11250 килобайт

**6.**   
из 2 сс в 10 сс в 8 сс

из 2 сс в 10 сс в 16 сс

***Билет номер 25***

**1.**   
Log2(30) = 4,91 = 5  
Log2(7) = 2,81 = 3  
5+3 = 8

**2.**

1.5кб = 1536 байтов

3072 / 1536 = 2 символа

**3.**

По форме представления :

Символьная, графическая, текстовая, звуковая, видео

По способу восприятия :

Визуальная, аудиальная, тактильная, вкусовая, обоятельная

По общественному значению :

Массовая, специальная, личная

**4.**

168,375 из 10 сс в 2 сс в 16 сс

0,1010 1000 011

1000000 + 1000 = 1001000

0100 1000    1010 1000     0110 0000    0000 0000

  4        8           A        8           6        0          0        016

**5.**

import math

def task1(n, m,):

   k = 1

   V = n \* m \* k                 #Находим объем памяти данного изображения

   return int (V)

if name == "main" :

   n = int (input("Ширина: "))

   m = int (input("Высота: "))

   print (task1(n, m)

**6.**

Механические, акустические, оптические и электрические

***Билет номер 26***

**1.**

Log2(67) = 6,07 бит

**2.**

1024 / 8 = 128 кб = 128 \* 1024 = 131072 байт = 131072 \* 8 = 1048576 бита

1048576 / 4096 = 256 бита на символ

4096 / 256 = 16 символов

**3.**

Процесс дискретизации информации заключается в преобразовании непрерывного сигнала в дискретный, то есть в представлении непрерывной функции в виде последовательности отдельных значений.

Аудиосигналы, изображения, видео, Данные о физических явлениях

**4.**

D9711 по двоично-шестнадцатеричной системе переводим каждое число отдельно и дополняем нулями

D – 1101 9 – 1001 7 – 0111 1 – 0001 1 – 0001

1101 1001 0111 0001 0001

1011001 – 1000000 = 11001

0111 000,1 0001 = - 56,53125

**5.**

def task3(nn, mm, vv):

    vv = vv \* 1024 \* 8                             #Перевод Килобайтов в биты

    k = vv / (nn \* mm)                             #определение кол-во цветов

    return int (k)

 nn = int (input("Ширина: "))

    mm = int (input("Высота: "))

    vv = int (input("Объем памати: "))

    print (task3(nn, mm, vv))

**6.**

Сиплекс, Дуплекс, Полудуплекс

***Билет номер 27***

**1.**

I = k \*\* i

N = 31 \* 12 \* 24 = 8928

I = log2(8928) = 13,12 бит

**2.**

8775 байт = 8775 \* 8 бит = 70200 байт

N = 2i

30 = 2i

i = 6

n = i / 6 \* 6 \* 30 = 70200 / 1080 = 65 символов в строке

**3.**

Разговор двух людей в природе, а в технике передача сообщений с другого пк на другой

**4.**

254,1285 из 10 сс в 2 сс в 16 сс

0.1111  1110  0010  0000  1110

1000000 + 1000 = 1001000

0100   1000   1111  1110  0010  0000  1110  0000

    4         8          F         E         2        0          E        016

**5.**

*def task4 (k, vvv):*

*vvv = vvv \* 8                                   #перевод из байтов в бит*

*k = math.log2 (k)                               #узнаём кол-во бит в одной точке*

*mn = vvv / k                                    #кол-во точек на экране*

*return int (mn)*

*k = int (input("кол-во цветов: "))*

*vvv = int (input("объем памяти: "))*

*print (task4(k, vvv))*

**6.**

аналоговые и дискретные

***Билет номер 28***

**1.**

N=2H

N=264

N=1.8\*1019

**2.**

0,0625 гб = 67108864 байтов

67108864 / 2048 = 32768 символа

**3.**

текстовая - буквы символы, графическая - изображение предметов графики, звуковая - устная речь

**4.**

E5612 по двоично-шестнадцатеричной системе переводим каждое число отдельно и дополняем нулями

E – 1110 5 – 0101 6 – 0110 1 – 0001 2 – 0010

1110 0101 0110 0001 0010

1100101 – 1000000 = 100101

0110 00,01 0010

24,28125

**5.**

def task3(v2, resolution2, rate2):

    v2 = v2 \* (1024 \*\* 3)                #Перевод гигабайтов в байты

    time2 = v2 / (rate2 \* resolution2 \* (8))   #Находим время звучания в секундах

    return float(time2)

v2 = float(input("объем в гигибайтов (писать через точку): "))

    rate2 =  int(input("частота: "))

    resolution2 = int(input("разрешение: "))

    print (task3(v2, rate2, resolution2))

перевод гб в байт, делим байты на частоту умноженную на разрядность на 8

**6.**

Коммутирумые ( временные ) , некоммутирумые ( выделеные )

***Билет номер 29***

**1.**

26=64 этажа

**2.**

2\*1,5\*1023= 3\*10\*23

**3.**

мера, характеризующая уменьшение неопределенности, содержащейся в одной случайной величине относительно другой

**4.**

113,475 из 10 сс в 2 сс в 16 сс

0.1110 0010 1111 0011 0010

1000000 + 0111 = 1000111

0100 0111 1110 0010 1111 0011 0010 0000

  4         7       E       2       F         3        2        016

**5.**

def task3(v2, resolution2, rate2):

    v2 = v2 \* 1024                 #Перевод килобайты в байты

    time2 = v2 / (rate2 \* resolution2 \* (8))  #Находим время звучания в секундах

    return float(time2)

v2 = float(input("объем в гигибайтов (писать через точку): "))

    rate2 =  int(input("частота: "))

    resolution2 = int(input("разрешение: "))

    print (task3(v2, rate2, resolution2))

перевод клб в байт, делим байты на частоту умноженную на разрядность на 8

**6.**

Модуля́ция — процесс изменения одного или нескольких параметров модулируемого несущего сигнала при помощи модулирующего сигнала.

***Билет номер 30***

**1.**

Log2(32)= 5 бит

**2.**

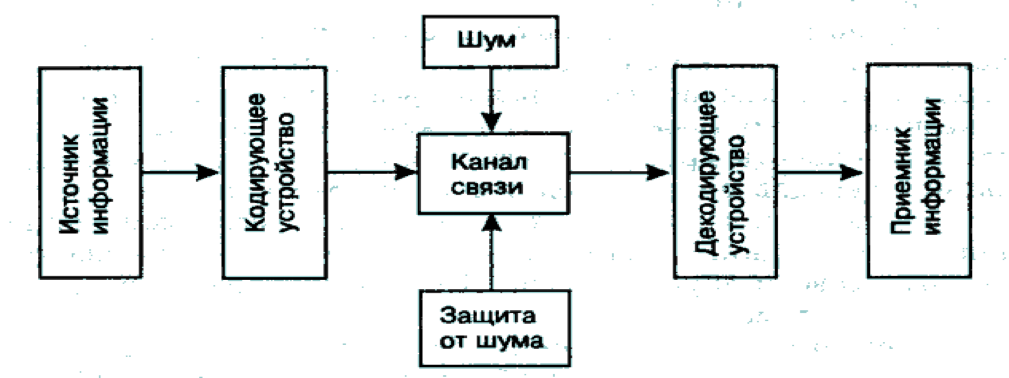
I = k \* i

1) I = 2 \* k

2) I = 7 \* k (128 символов)

3) 7 / 2 = 3.5

**3.**



**4.**

F9811 по двоично-шестнадтцатиричной системе переводим каждое число отдельно и дополняем нулями

F- 1111 9- 1001 8-1000 1-0001  1-0001

1 111 1001 1000 0001 0001

1111 001-1000000 = 11101

1000 000,1 0001 = - 64,5312510

**5.**

3 \* 60 = 180 секунд  
16 \* 1000 = 16000  
180 \* 16000 \* 32 / 8 = 11520000 байт   
11520000 / 1024 = 11250 килобайт

**6.**

это набор правил и соглашений, которые определяют формат и порядок обмена информацией между устройствами в сети.

***Билет номер 31***

**1.**

Log2(25) = 5   
Log2(8)  = 3  
5+3 = 8

**2.**

3кб = 3072 байтов

3072 / 3072 = 1 символа

**3.**

Информация — это сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые воспринимают информационные системы в процессе жизнедеятельности и работы.

Данные — это сведения, факты, показатели, выраженные как в числовой, так и в любой другой форме.

Знания — это совокупность сведений (данных или программ), отражающих знания человека — специалиста (эксперта) в определенной предметной области. Знания предназначены для хранения в базах знаний

**4.**

178,325 из 10 сс в 2 сс в 16 сс

0.1011 0010 0101 0011 0011

1000000 + 1000 = 1001000

0100 1000 1011 0010 0101 0011 0011 0000

   4        8     B        2         5       3        3        016

**5.**

*def task4 (k, vvv):*

*vvv = vvv \* 8                                   #перевод из байтов в бит*

*k = math.log2 (k)                               #узнаём кол-во бит в одной точке*

*mn = vvv / k                                    #кол-во точек на экране*

*return int (mn)*

*k = int (input("кол-во цветов: "))*

*vvv = int (input("объем памяти: "))*

*print (task4(k, vvv))*

**6.**

Основание системы счисления — это количество цифр или других символов, необходимых для написания числа в этой системе счисления.

***Билет номер 32***

**1.**

Log2(63) = 5.98 бит

**2.**

0.25 мб = 2097152 бит

2097152 / 2018 = 1039,223 символа

**3.**

неотрывность информации от физического носителя и языковая природа информации.

**4.**

D9721 по двоично-шестнадтцатиричной системе переводим каждое число отдельно и дополняем нулями

D- 1101 9- 1001 7-0111 2-0010  1-0001

1101 1001 0111 0010 0001

1011001-1000000 = 11001

0111 001.0 0001 = -57.03125

**5.**

def task3(nn, mm, vv):

    vv = vv \* 1024 \* 8                             #Перевод Килобайтов в биты

    k = vv / (nn \* mm)                             #определение кол-во цветов

    return int (k)

 nn = int (input("Ширина: "))

    mm = int (input("Высота: "))

    vv = int (input("Объем памати: "))

    print (task3(nn, mm, vv))

**6.**

Чтобы перевести дробное число из десятичной системы счисления в любую позиционную, нужно последовательно умножать дробную часть числа на основание системы счисления, в которую переводится число, и выделять целые части получающихся произведений до тех пор, пока дробная часть не станет равна нулю или пока не будет достигнута нужная точность. Затем нужно последовательно приписать к целой части полученные ранее целые части произведений, начиная с последнего.