

## Рабочая тетрадь № 1

«Информатика – это наука о методах и процессах сбора, хранения, обработки, передачи, анализа и оценки информации с применением компьютерных технологий, обеспечивающих возможность ее использования для принятия решений».

*Большая российская энциклопедия, 2008.*

«Информация – это сведения, независимо от формы их представления, воспринимаемые человеком или специальными устройствами как отражение фактов материального мира в процессе коммуникации».

*ГОСТ 7.0-99.*

### 1. Теоретический материал

Бит – это минимальная единица измерения информации. Бит может принимать только два значения 0 или 1 [1, 2].

В вычислительных системах приняты следующие единицы измерения представления цифровых данных:

8 бит = 1 байт (1 Б),

1024 байт (1024 Б) = 1 килобайт (1 Кбайт),

1024 килобайт (1024 Кбайт) = 1 мегабайт (1 Мбайт),

1024 мегабайт (1024 Мбайт) = 1 гигабайт (1 Гбайт),

1024 гигабайт (1024 Гбайт) = 1 терабайт (1 Тбайт).

### 2. Пример

**Задача:**

Сколько битов в 3 Мбайт?



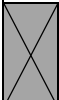
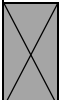
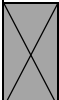
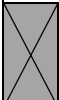

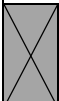
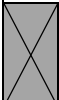
**Решение:**

$3 \text{ Мбайт} = 3 * 1024 \text{ Кбайт} = 3 * 1024 * 1024 \text{ Б} =$   
 $= 3 * 1024 * 1024 * 8 \text{ бит} = 25165824 \text{ бит}$

**Ответ:**

25165824 бит

### 3. Задания

1.	<b>Задача:</b>
	Сколько битов в 4 Мбайт?
	<b>Решение:</b>
	$4 \cdot 1024 \cdot 1024 \cdot 8$
	<b>Ответ:</b>
	<input type="text"/>
2.	<b>Задача:</b>
	Сколько байтов в 2 Гбайт?
	<b>Решение:</b>
	$2 \cdot 1024 \cdot 1024 \cdot 1024$
	<b>Ответ:</b>
	<input type="text"/>
3.	<b>Задача:</b>
	Переведите 6291456 байт в Мбайт?
	<b>Решение:</b>
	$6291456 / 1024 / 1024$
	<b>Ответ:</b>
	<input type="text"/>

### 1. Теоретический материал

Формула  $N = \log_2 K$ , где  $K$  – количество возможных состояний, а  $N$  – минимальное количество информации в битах, необходимое для описания состояний системы – формула Хартли [1, 2]. При вычислении по формуле Хартли может получено нецелое значение  $N$ . В этом случае значение необходимо округлить вверх до целого значения.

## 2. Пример

### 1. Задача:

Сколько бит нужно отвести на кодирование букв русского алфавита, если **НЕ** различать буквы Е и Ё?

А	Б	В	Г	Д
Е	Ё	Ж	З	И
Й	К	Л	М	Н
О	П	Р	С	Т
У	Ф	Х	Ц	Ч
Ш	Щ	Ъ	Ы	Ь
Э	Ю	Я		

### Решение:

Если не различать буквы Е и Ё, то в русском алфавите 32 буквы. Тогда:  $\log_2 32 = 5$ . Получилось целое число.

### Ответ:

5 бит

### 2. Задача:

Сколько бит нужно отвести на кодирование букв русского алфавита, если различать буквы Е и Ё?

### Решение:

Если различать буквы Е и Ё, то в русском алфавите 33 буквы. Тогда:  $\log_2 33 = 5,044$ . Получилось не целое число. Поэтому округлим в верхнюю сторону до 6.

### Ответ:

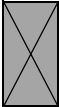
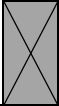
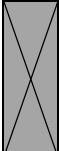
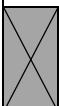
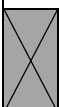
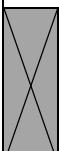
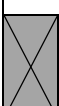
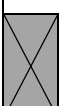
6 бит

## 3. Задания

### 1. Задача:

Сколько бит нужно отвести на кодирование букв английского алфавита?

A B C D E F G H  
I J K L M N O P  
Q R S T U V W X  
Y Z

	<b>Решение:</b>
	 $\log_2(26) = \sim 4.7 \Rightarrow 5$
	<b>Ответ:</b>
	 <input type="text"/>
2.	<b>Задача:</b>
	 Сколько бит нужно отвести на кодирование гласных букв английского алфавита?
	<b>Решение:</b>
	 $\log_2(5) = \sim 2.3 \Rightarrow 3$
	<b>Ответ:</b>
	 <input type="text"/>
3.	<b>Задача:</b>
	 Сколько бит нужно отвести на кодирование согласных букв английского алфавита?
	<b>Решение:</b>
	 $\log_2(21) = \sim 4.3 \Rightarrow 5$
	<b>Ответ:</b>
	 <input type="text"/>

## 1. Теоретический материал

Пусть теперь система может находиться в одном из  $K$  состояний с разными вероятностями [1]. В состоянии 1 с вероятностью  $p_1$ , в состоянии 2 с вероятностью  $p_2$  и продолжая рассуждения в состоянии  $K$  с вероятностью  $p_k$ , где  $p_k \geq 0$ . Тогда ценность знания, что система находится в состоянии  $p_k$  зависит от распределения вероятностей [1].

Фундаментальное понятие теории информации – энтропия информации. Под энтропией понимается мера неопределенности системы. Энтропией по Шеннону называется число

$$H = - \sum_{i=1}^N p_i \log_2(p_i).$$

Прирост информации – это уменьшение энтропии.

## 2. Пример

### Задача:

Если двоечник не поступил в РТУ МИРЭА, то тут мало информации, потому что «мы это и так знали», а вот если он поступил, то это «новость»!

Полагая, что двоечник не поступает с вероятностью 0,9, а поступает с вероятностью 0,1, найдите энтропию по Шеннону [1].

### Решение:

Частная энтропия для не поступления равна:

$$-0,9 * \log_2 0,9 = 0,137,$$

а для поступления равна:

$$-0,1 * \log_2 0,1 = 0,332.$$

А общая энтропия равна:

$$H = 0,137 + 0,332 = 0,469.$$

### Ответ:

$$H = 0,469.$$

## 3. Задания

### 1. Задача:

Найти энтропию подбрасывания одной монеты.

### Решение:

$$2 * (-0.5 * \log_2(0.5))$$

	<b>Ответ:</b>
	1
2.	<b>Задача:</b>
	Найти энтропию подбрасывания игральной кости.
	<b>Решение:</b>
	$\text{Sigma}(n=1, 6, -(1/6) \cdot \log_2(1/6)) = \sim 2.58$
	<b>Ответ:</b>
	2.58
3.	<b>Задача:</b>
	Имеется очень загруженный сервер. Из-за этого с вероятностью $p_1 = 0,6$ сервер принимает запрос на обработку данных и с вероятностью $p_2 = 0,4$ отвергает его. Найти частные энтропии и общую энтропию системы.
	<b>Решение:</b>
	$\text{pr1} = -0.6 \cdot \log_2(0.6) = \sim 0.44$ $\text{pr} = \text{pr1} + \text{pr2} = \sim 0.99$ $\text{pr2} = -0.4 \cdot \log_2(0.4) = \sim 0.52$
	<b>Ответ:</b>
	0.97

## 1. Теоретический материал

Кодирование – это процесс преобразования данных в цифровой формат для хранения, передачи и обработки в вычислительных системах.

Вся цифровая информация в вычислительных системах представляется в двоичном коде – наборе нулей и единиц. Двоичный код – это кодирование каждого объекта последовательностью бит.

На рисунке представлена таблица ASCII кодов символов английского алфавита с учетом прописных букв, знаков препинания, чисел, арифметических операций и некоторых других вспомогательных символов.

Таблица ASCII однозначно определяет 128 символов, расширенная ASCII таблица содержит 256 символов (1 байт) и включает русский алфавит.

**ASCII Code Chart**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

При кодировании символов всех алфавитов (японского, китайского и других) одного байта недостаточно, поэтому применяется Unicode, который измеряется двумя байтами – количество символов  $2^{16} = 65536$ .

## 2. Пример

**Задача:**

Найти ASCII код символа **N**

**Решение:**

Символ **N** находится на пересечении строки **4** и столбца **E**, поэтому он кодируется **4E<sub>16</sub>** в шестнадцатеричной системе счисления.

**Ответ:**

**4E<sub>16</sub>**

## 3. Задания

1. **Задача:**

Найти ASCII код символа **w** (строчная буква).

**Решение:**

w 7 7 => w = 0x77

**Ответ:**

0x77

2. **Задача:**

Найти символ по ASCII коду **75<sub>16</sub>**.

X	<b>Решение:</b>
X	5 7 u => 0x75 = u
X	<b>Ответ:</b>
X	u

3.	<b>Задача:</b> Запишите слово <b>student</b> набором символов ASCII кода.
X	<b>Решение:</b>
X	s = 0x73; t = 0x74; u = 0x75; d = 0x64; e = 0x65; n = 0x6E => student = 0x73,0x74,0x75,0x64,0x65,0x6E,0x74
X	<b>Ответ:</b>
X	0x73,0x74,0x75,0x64,0x65,0x6E,0x74

## 1. Теоретический материал

Важная задача кодирования – это возможность обнаружения ошибок, которые возникают в процессе хранения и/или передачи информации [1].

Рассмотрим простейший способ обнаружения однократной ошибки – бит четности. Например, байт представлен восьмью битами. Тогда в пересылаемом сообщении добавляется девятый бит – бит четности, который равен единице, если количество бит в исходном байте нечетно и нулю, если количество бит четно.

Исходное сообщение	Пересылаемое сообщение	Полученное сообщение	Проверка
01101101	01101101 <b>1</b>	01101101 <b>1</b>	Ок. Чётное число единиц
01010101	01010101 <b>0</b>	01010101 <b>0</b>	Ок. Чётное число единиц
01011101	01011101 <b>1</b>	010 <b>0</b> 1101 <b>1</b>	Ошибка. Нечётное число единиц
01011101	01011101 <b>1</b>	01011101 <b>0</b>	Ошибка. Нечётное число единиц



## 2. Пример

### Задача:

Добавьте бит чётности к следующему сообщению: 01101011 .

### Решение:

В сообщении не четное число бит, поэтому в конце к нему нужно дописать единицу, чтобы пересылаемое сообщение содержало чётное число единиц, т.е. 011010111

### Ответ:

011010111

## 3. Задания

### 1. Задача:

Добавьте бит чётности к следующему сообщению: 11001001 .

### Решение:

11001001 1 => = 0

### Ответ:

110010010

### 2. Задача:

Добавьте бит чётности к следующему сообщению: 10100001 .

### Решение:

10100001 1 => = 1

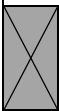
### Ответ:

101000011

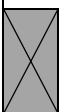
### 3. Задача:

Было принято следующее сообщение: 101011010 . Содержит ли оно ошибку?

**Решение:**



**Ответ:**

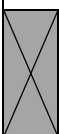
☐

## 1. Теоретический материал

Рассмотрим еще подход к обнаружению ошибки – троированию бита в передаваемом сообщении. Пусть имеется один байт 10010010, тогда при передаче сообщения каждый бит будет троирован и сообщение примет вид (111)(000)(000)(111)(000)(000)(111)(000). Скобки в примере применены для наглядности представления записи. Каждая скобка соответствует одному биту исходного сообщения закодированного по методу троирования. Искажение одного бита в скобке позволит выявить возникшие ошибки (101)=1, (100)=0.

## 2. Пример

1. **Задача:**



Используя кодирование с избытком, закодируйте следующее сообщение: 10101101, троированием битов.

**Решение:**



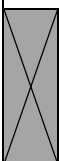
| 111 | 000 | 111 | 000 | 111 | 111 | 000 | 111 |

**Ответ:**



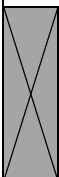
11100011100011111000111

2. **Задача:**



В предположении, что в трех идущих подряд битах не может быть более одной ошибки, восстановите следующее сообщение: 101000111001111101001111 .

**Решение:**



до	101	000	111	001	111	101	001	111
после	111	000	111	000	111	111	000	111
результат	1	0	1	0	1	1	0	1

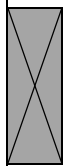
**Ответ:**



10101101

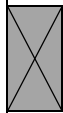
### 3. Задания

1. **Задача:**



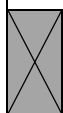
Используя кодирование с избытком, закодируйте следующее сообщение:  
00111011, троированием битов.

**Решение:**



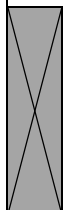
000 | 000 | 111 | 111 | 111 | 000 | 111 | 111

**Ответ:**



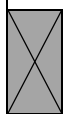
0000001111111111000111111

2. **Задача:**



В предположении, что в трех идущих подряд битах не может быть более одной ошибки, восстановите следующее сообщение:  
001011010110011010001111 .

**Решение:**

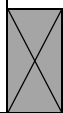


001 | 011 | 010 | 110 | 011 | 010 | 001 | 111

000 | 111 | 000 | 111 | 111 | 000 | 000 | 111

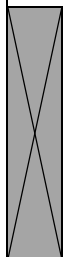
=> = 01011001

**Ответ:**



01011001

3. **Задача:**



В сообщении троировались байты (символы таблицы ASCII). Было получено следующее сообщение:

CCzoYomdmppSuRutptweeQrr\_\_\*RssacciBieeen%Fccjee .

Восстановите исходное сообщение.

**Решение:**

CCz | oYo | mdm | ppS | uRu | tpt | wee | Qrr | \_\* | Rss | acc | iBi | eee | nn% | Fcc | jee  
 CCC | ooo | mmm | ppp | uuu | ttt | eee | rrr | \_ | sss | ccc | iii | eee | nnn | ccc | eee

= Computer\_science

**Ответ:**

Computer\_science

**1. Теоретический материал**

Определим расстояние между символами кода. Пусть каждый символ кодируется последовательностью из  $N$  битов  $x = (x_1, x_2, \dots, x_N)$ ,  $y = (y_1, y_2, \dots, y_N)$ . Расстояние  $\rho(x, y)$  определим следующей формулой [1]:

$$\rho(x, y) = \sum_{i=1}^N |x_i - y_i|,$$

где  $x_i$  и  $y_i$  принимают значения ноль или единица. Представленная формула позволяет определить расстояние между представлениями символов в виде кода. Тогда  $n$  количество ошибок, которые можно исправить, если определить наименьшее из расстояний  $d$  определяется по следующей формуле:

$$d \geq 2n + 1.$$

Обобщим, идея метода состоит в определении количество отличных битов в кодовом представлении символов. Далее из всех выбирается наименьшее значение – это значение с использованием формулы  $d \geq 2n + 1$ , позволяет найти количество ошибок, которые можно исправить. То есть, чтобы исправлять  $n$  ошибок, необходимо учитывать расстояние между любыми символами, которое должно быть не меньше  $2n + 1$ .

**2. Пример****1. Задача:**

Даны следующие коды:

буква	A	B	C
код	00000	11100	00111

Найти расстояние между кодами для **B** и **C**.

**Решение:**

Посчитаем количество различных бит в кодах для **B** и **C**. Четыре бита различны.

B	1	1	1	0	0
C	0	0	1	1	1

**Ответ:**

$$\rho(B, C) = \rho(11100, 00111) = 4$$

2. **Задача:**

Сколько ошибок можно исправить при использовании кодов из предыдущего примера?

**Решение:**

Найдём минимальное расстояние  $d$  между кодами.

A	0	0	0	0	0
B	1	1	1	0	0
$\rho(A, B) = 3$					

B	1	1	1	0	0
C	0	0	1	1	1
$\rho(B, C) = 4$					

C	0	0	1	1	1
A	0	0	0	0	0
$\rho(C, A) = 3$					

$$d = \min(3, 4, 3) = 3$$

$$d \geq 2n + 1 \quad \rightarrow \quad 3 \geq 2n + 1$$

Из этого условия найдём  $n$ . Получим  $n = 1$ . Таким образом, всегда можно исправить 1 ошибку.

**Ответ:**

Можно гарантированно исправить 1 ошибку

### 3. Задания

1. **Задача:**

Даны следующие коды:

буква	A	B	C
код	0101010	1010100	0011110

Найти расстояние между кодами для **B** и **C**.

**Решение:**

$$\rho(B, C) = 3 \text{ ( : 0, 3, 5 ( , ))}$$

**Ответ:**

$$\rho(B, C) = 3$$

2.	<p><b>Задача:</b></p> <p>Верны ли следующие утверждения:</p> $\rho(A, B) \leq \rho(A, C) + \rho(C, B) ,$ $\rho(A, C) \leq \rho(A, B) + \rho(B, C) ,$ $\rho(B, C) \leq \rho(B, A) + \rho(A, C) ?$
	<p><b>Решение:</b></p> <div> <div> <math>\rho(A, B) = 6; \rho(A, C) = 3; \rho(B, C) = 3</math> </div> <div> <math>6 \leq 3+3: \text{True}</math>  <math>3 \leq 6+3: \text{False}</math>  <math>3 \leq 6+3: \text{False}</math> </div> </div>
	<p><b>Ответ:</b></p> <p>1. True; 2. False; 3. False</p>
3.	<p><b>Задача:</b></p> <p>Сколько ошибок можно гарантированно исправить при использовании кодов из задачи 1?</p>
	<p><b>Решение:</b></p> <div> <math>d = \min(6, 3, 3) = 3 \Rightarrow 3 \geq 2n+1 \Rightarrow n = 1</math> </div>
	<p><b>Ответ:</b></p> <div></div>

Тест 1	
1.	<p>Сколько бит в 1 Кбайт?</p> <div> <div>1. 8192 бит</div> <div>2. 75699 бит</div> <div>3. 4589 бит</div> <div>4. 34773 бит</div> </div>
	<p><b>Ответ:</b></p> <div></div>
2.	<p>Сколько бит нужно отвести на кодирование одной игровой карты стандартной колоды из 36 карт?</p> <div> <div>1. 4</div> <div>2. 5</div> <div>3. 6</div> <div>4. 7</div> </div>
	<p><b>Ответ:</b></p> <div></div>

3.	Сколько бит нужно отвести на кодирование двузначного десятичного числа?
<input checked="" type="checkbox"/>	1. 6                                      2. 7                                      3. 8                                      4. 9
	<b>Ответ:</b>
<input checked="" type="checkbox"/>	<input type="text"/>
4.	Если не прогуливать занятия, то вероятность сдать сессии на «хорошо» и «отлично» равна 0,7. Найдите энтропию системы.
<input checked="" type="checkbox"/>	1. -0,05                                      2. 0,91                                      3. 0,13                                      4. 0,88
	<b>Ответ:</b>
<input checked="" type="checkbox"/>	(4) 0,88
5.	Укажите ASCII код символа G
<input checked="" type="checkbox"/>	1. 67                                      2. 43                                      3. 6A                                      4. 47
	<b>Ответ:</b>
<input checked="" type="checkbox"/>	(4) 0x47
6.	Бит четности служит для
<input checked="" type="checkbox"/>	1. исправления ошибок в данных 2. обнаружения ошибки в данных 3. шифрования данных 4. выравнивания данных
	<b>Ответ:</b>
<input checked="" type="checkbox"/>	<input type="text"/>
7.	Если при пересылке сообщения в нём произошло ДВЕ ошибки, то бит четности
<input checked="" type="checkbox"/>	1. позволит исправить две ошибки 2. позволит их обнаружить 3. не позволит их обнаружить 4. позволит исправить только одну ошибку
	<b>Ответ:</b>
<input checked="" type="checkbox"/>	<input type="text"/>
8.	В предположении, что в трех идущих подряд битах не может быть более одной ошибки, восстановите следующее сообщение: 001011101010100000001110 .
<input checked="" type="checkbox"/>	1. 01100001                                      2. 01101101 3. 11100001                                      4. 01000011

	<b>Ответ:</b>
	<input type="text" value="(1) 01100001"/>
<b>9.</b>	Найдите, между какими кодами расстояние наибольшее
	1. 11100010 и 00001111 2. 00010111 и 01110101 3. 01011010 и 10110101 4. 10101010 и 10101101
	<b>Ответ:</b>
	<input type="text" value="(2) 01011010 10110101 (= 7)"/>
<b>10.</b>	Иконка на рабочем столе имеет разрешение 32x32 пикселя. На кодирование каждого пикселя отводится 24 бита. Найдите сколько бит нужно отвести на кодирование одной иконки.
	1. 54576 бит 2. 32679 бит 3. 16384 бит 4. 24576 бит
	<b>Ответ:</b>
	<input type="text"/>

## Реализация задач на языке программирования Python

Для реализации задач необходимо установить интерпретатор языка Python. Среду разработки и интерпретатор можно бесплатно установить с официального сайта [www.python.org](http://www.python.org). Также, можно бесплатно установить среду разработок Anaconda с сайта <https://www.anaconda.com/products/individual>. Однако, для начального ознакомления с синтаксисом языка можно использовать онлайн интерпретаторы, например, <https://www.online-python.com>.

### 1. Теоретический материал

Давайте создадим первую программу на Python.

```
print('Hello world!')
```

Функция `print()` выводит на экран сообщение в скобках. Кавычки окаймляют текст 'Hello world!'.



Функция `input()` используется для ввода данных с клавиатуры:

```
name = input('Введите имя')  
print('Привет, ' + name)
```

Здесь `name` – имя переменной. Имена переменных используются для хранения значений. Символ `+` используется для соединения (конкатенации) строк.

Python содержит все необходимые математические операции.

```
print(5 + 7)    # сложение  
print(4 * 5)    # умножение  
print(4 ** 3)   # возведение в степень
```

После символа `#` записываются комментарии, которые игнорируются интерпретатором.

## 2. Пример

**Задача:**

Найти значение функции  $f(x) = x^2 + 3x - 100$ . Значение  $x$  вводится с клавиатуры.

**Решение (код программы):**

```
x = input('Введите x')    # возвращается строка, не число  
x=float(x)                # преобразуем строку в вещественное число  
y=x**2+3*x-100  
print(y)
```





## 3. Задания

1. **Задача:**

Выведите на экран вашу Фамилию, Имя и номер студенческой группы.

**Решение (код программы):**

```
print(" ; -01-24")
```

2.	<b>Задача:</b>
	Введите с клавиатуры два числа и сложите их. Выведите результат на экран.
	<b>Решение (код программы):</b>
	<code>print(sum([int(i) for i in input().split(" ")])</code>
3.	<b>Задача:</b>
	Найти значение функции $f(x) = x^5 - 2x^3 + 1$ . Значение $x$ вводится с клавиатуры.
	<b>Решение (код программы):</b>
	<code>x = int(input("x: ")); print("f(x) = {0}".format((x**5) - 2*(x**3) + 1))</code>

## 1. Теоретический материал

Примеры различных типов данных:

```
_string = 'строка'    # строка
_integer = 12          # целое число
_float_1 = 3.14        # вещественное число
_float_2 = -2.7e-3     # -0.0027
_boolean = True        # False
```

Тип переменной всегда можно узнать с помощью функции **type()**

```
print( type(_boolean) )    # <class 'bool'>
```

В Python есть следующие операции сравнения: **==** (проверка на равенство), **!=**(не равняется), **<** , **<=**(меньше или равняется), **>**, **>=**

```
print(2+1 > 3*4)          # False
```

В Python есть следующие логические операции: **and**(логическое И),**or**(логическое ИЛИ), **not**(логическое отрицание).

```
print( not (3>1 and False) )    # True
```

В Python есть также тип **list** (список), который позволяет хранить совокупность различных объектов:

```
empty_list = []                # пустой список
_list = [1, 3.14, 'свет', True, []] # список с элементами
empty_list.append( 12 )         # добавление элемента
empty_list.append( [2.7, 3] )
```

```
print( empty_list, _list ) #
_list[0] = 'перезаписываем первый элемент на этот текст'
print( _list, empty_list[1] )
```

## 2. Пример

1. **Задача:**

Проверить тип результата сложения целого числа с вещественным.

**Решение (код программы):**

```
a = 12 + 3.14
print( type(a) ) # функция type возвращает тип её аргумента
```

2. **Задача:**

Определите истинность следующего выражения:

$$\frac{9}{3} > 2 * 3 \text{ or } \neg(12 \neq 3^2 + 3 \text{ and } 57 - 24 > 30)$$

**Решение (код программы):**

```
print(9/3 > 2*3 or not(12 != 3**2+3 and 57-24 > 30) )
```

## 3. Задания

1. **Задача:**

Напишите код для определения типа переменной **strange**, если:

```
strange = [ [],1]
```

**Решение (код программы):**

```
strange = [[],1]; print(type(strange))
```

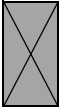

2. **Задача:**

С помощью Python найдите такие значения  $x$  и  $y$ , которые обратят выражение в значение True.

Выражение:  $(x \text{ or } y) \text{ and } (\neg x \text{ or } y) \text{ and } \neg(x \text{ and } y)$ .

**Решение (код программы):**

```
res = []for x in range(2):for y in range(2):if((x or y) and (not x or y) and not(x and y)):res.append((x,y))
```

3.	<b>Задача:</b>
	Добавьте в пустой список четыре любых значения и выведите их на экран в обратном порядке, используя для этого индексы элементов.
	<b>Решение (код программы):</b>
	<pre>s = [] for i in range(4): s.append(i) print(s[::-1])</pre>

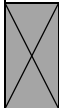
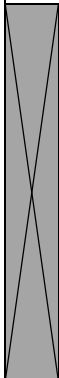

## 1. Теоретический материал

Язык Python включает в себя множество полезных библиотек. Библиотека **math** является одной из таких. Она содержит все стандартные математические функции. Для использования библиотеку необходимо подключить:

```
import math as m

a    = m.sin(m.pi/2)      #  $\sin\left(\frac{\pi}{2}\right)$ 
b    = m.sqrt( 16 )       #  $\sqrt{16}$ 
c_1  = m.e**2             #  $e^2$ 
c_2  = m.exp(2)           #  $e^2$ 
d_1  = m.log(8, 2)        #  $\log_2(8)$ 
d_2  = m.log2(8)          #  $\log_2(8)$ 
e_1  = m.ceil(3.14)       # округление вверх (ответ 4)
e_2  = m.ceil(2.7)        # округление вверх (ответ 3)
```

## 2. Пример

1.	<b>Задача:</b>
	Написать программу для решения квадратного уравнения, через дискриминант: $3x^2 - 10x + 1 = 0$ .
	<b>Решение (код программы):</b>
	<pre>import math as m a, b, c = 3, -10, 1 D      = b**2-4*a*c x_1    = (-b-m.sqrt(D))/(2*a) x_2    = (-b+m.sqrt(D))/(2*a) print(x1, x2)</pre>
2.	<b>Задача:</b>
	Напишите программу для вычисления $\log_2(7 * x) * \cos\left(\frac{x}{3}\right)$ , где $x$ вводит

	пользователь с клавиатуры.
	<b>Решение (код программы):</b>
	<pre>import math as m x = float( input("Введите x: ") ) print( m.log2(7*x)*m.cos(x/3) )</pre>

### 3. Задания

1.	<b>Задача:</b>
	Запрограммируйте формулы Хартли. Количество состояний вводится с клавиатуры.
	<b>Решение (код программы):</b>
	<pre>import math; print(int(math.ceil(math.log(int(input("n: ")),2))))</pre>
2.	<b>Задача:</b>
	Запрограммируйте вычисление энтропии по Шеннону для систем из двух состояний. Вероятности вводятся с клавиатуры.
	<b>Решение (код программы):</b>
	<pre></pre>
3.	<b>Задача:</b>
	Напишите программу для вычисления $\left\{ \tan \left( \frac{\cos(x) \cdot \sin(2x)}{x \cdot e^x} \right) \right\}^{\log_7(x)}$ , где $x$ вводит пользователь с клавиатуры.
	<b>Решение (код программы):</b>
	<pre>res = m.tan((m.cos(x)*m.sin(2*x))/(x*(m.e**x)))**\ m.log(x,7)</pre>
4.	<b>Задача:</b>
	Напишите программу для добавления бита четности к байту. Байт можете записать в виде списка (list) нулей и единиц.
	<b>Решение (код программы):</b>
	<pre>b = [int(i) for i in str(input("byte (00010100): "))] b.append(b.count(1)%2) print(b) print("".join([str(i) for i in b]))</pre>

```
import math
prob = [float(i) for i in input("probabilities (0.7 0.3): ").split(" ")]
print("H: {0}".format(sum([-prob[i]*math.log(prob[i],2) for i in range(len(prob))])))
```