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 Мбайт = 3 \* 1024 Кбайт = 3\* 1024 \* 1024 Б =  = 3\* 1024 \* 1024 \* 8 бит = 25165824 бит |
| ***Ответ:*** | |
|  | 25165824 бит |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Сколько битов в 4 Мбайт? |
| ***Решение:*** | |
|  | 4 Мбайт = 4 \* 1024 Кбайт = 4\* 1024 \* 1024 Б =  = 4\* 1024 \* 1024 \* 8 бит = 33554432 бит |
| ***Ответ:*** | |
|  | 33554432 бит |
| 2. | ***Задача:*** | |
|  | Сколько байтов в 2 Гбайт? |
| ***Решение:*** | |
|  | 2 \*1024 \*1024 \*1024 =2147483648 байт |
| ***Ответ:*** | |
|  | 2147483648 байт |
| 3. | ***Задача:*** | |
|  | Переведите 6291456 байт в Мбайт? |
| ***Решение:*** | |
|  | 6291456/1024/1024= 6 Мбайт |
| ***Ответ:*** | |
|  | 6 Мбайт |
|  |  |  |

|  |
| --- |
| **1. Теоретический материал** |
| Формула N = log2K, 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? |  | |
| ***Решение:*** | |
|  | Если не различать буквы Е и Ё, то в русском алфавите 32 буквы. Тогда: log232= 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: log233 = 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? | Описание: C:\Users\Qverty\Temp PPS\26_Информатика\Рабочии тетради\английский алфавит 2.png | |

|  |  |  |
| --- | --- | --- |
|  | ***Решение:*** | |
|  | 26 букв Log226 = 5 бит (округ до целого) |
| ***Ответ:*** | |
|  | 5 бит |
| 2. | ***Задача:*** | |
|  | Сколько бит нужно отвести на кодирование гласных букв английского алфавита? |
| ***Решение:*** | |
|  | 5 гл букв. Log25 = 3 бит (округ до целого) |
| ***Ответ:*** | |
|  | 3 бита |
| 3. | ***Задача:*** | |
|  | Сколько бит нужно отвести на кодирование согласных букв английского алфавита? |
| ***Решение:*** | |
|  | 21 гл буквLog221 = 5 бит (округ до целого) |
| ***Ответ:*** | |
|  | 5 бит |

|  |  |
| --- | --- |
| **1. Теоретический материал** | |
| Пусть теперь система может находиться в одном из *K* состояний с разными вероятностями [1]. В состоянии 1 с вероятностью , в состоянии 2 с вероятностью и продолжая рассуждения в состоянии *K* с вероятностью , где . Тогда ценность знания, что система находится в состоянии зависит от распределения вероятностей [1].  Фундаментальное понятие теории информации – энтропия информации. Под энтропией понимается мера неопределенности системы. Энтропией по Шеннону называется число  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 \* log20,9 = 0,137,  а для поступления равна:  -0,1 \* log20,1 = 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:*** | |
|  | Выпадение одной из сторон -0,5 \* log20,5 = 0,5  Общая H=0,5+0,5=1 |

|  |  |  |
| --- | --- | --- |
|  | ***Ответ:*** | |
|  | Н = 1 |
| 2. | ***Задача:*** | |
|  | Найти энтропию подбрасывания игральной кости. |
| ***Решение:*** | |
|  | -(1/6)\*log2(1/6) =0,43  0,43\*6=2,58 |
| ***Ответ:*** | |
|  | 2,58 |
| 3. | ***Задача:*** | |
|  | Имеется очень загруженный сервер. Из-за этого с вероятностью сервер принимает запрос на обработку данных и с вероятностью отвергает его. Найти частные энтропии и общую энтропию системы. |
| ***Решение:*** | |
|  | Частная принятия: -0.6\*log20.6=0.442  Частная отвержения -0.4\*log20.4=0.52  Общая энтропия = 0.442+0,52=0.962 |
| ***Ответ:*** | |
|  | Частная принятия -0.442  Частная отвержения 0.52  Общая энтропия 0.962 |

|  |  |
| --- | --- |
| **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.    При кодировании символов всех алфавитов (японского, китайского и других) одного байта недостаточно, поэтому применяется Unicode, который измеряется двумя байтами – количество символов 216 = 65536. | |
| **2. Пример** | |
| ***Задача:*** | |
|  | Найти ASCII код символа **N** |
| ***Решение:*** | |
|  | Символ **N** находится на пересечении строки **4** и столбца **E**, поэтому он кодируется **4E16** в шестнадцатеричной системе счисления. |
| ***Ответ:*** | |
|  | **4E16** |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Найти ASCII код символа **w** (строчная буква). |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 77**16** |
| 2. | ***Задача:*** | |
|  | Найти символ по ASCII коду **7516**. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | U |
| 3. | ***Задача:*** | |
|  | Запишите слово **student** набором символов ASCII кода. |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | 73 747564 65 6E74 (16) |

|  |
| --- |
| **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**1** | 01101101**1** | Ок. Чётное число единиц | | 01010101 | 01010101**0** | 01010101**0** | Ок. Чётное число единиц | | 01011101 | 01011101**1** | 01001101**1** | Ошибка. Нечётное число единиц | | 01011101 | 01011101**1** | 01011101**0** | 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. 01101011**1** |
| ***Ответ:*** | |
|  | 01101011**1** |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Добавьте бит чётности к следующему сообщению: 11001001 . |
| ***Решение:*** | |
|  | четное число бит, дописываем 0 |
| ***Ответ:*** | |
|  | 110010010 |
| 2. | ***Задача:*** | |
|  | Добавьте бит чётности к следующему сообщению: 10100001 . |
| ***Решение:*** | |
|  | нечетное число единиц, дописываем 1 |
| ***Ответ:*** | |
|  | 101000011 |
| 3. | ***Задача:*** | |
|  | Было принято следующее сообщение: 10101101**0** . Содержит ли оно ошибку? |
| ***Решение:*** | |
|  | Да так как бит четности оставляет число единиц нечетным |
| ***Ответ:*** | |
|  | да |

|  |  |  |
| --- | --- | --- |
| **1. Теоретический материал** | | |
| Рассмотрим еще подход к обнаружению ошибки – троировании 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:*** | |
|  | 111000111000111111000111 |
| 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:*** | |
|  | 000 000 111 111 111 000 111 111 |
| 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 => 0 1 0 1 1 0 0 1 |
| ***Ответ:*** | |
|  | 01011001 |
| 3. | ***Задача:*** | |
|  | В сообщении троировались байты (символы таблицы ASCII). Было получено следующее сообщение:  CCzoYomdmppSuRutptweeQrr\_\_\*RssacciBieeenn%Fccjee .  Восстановите исходное сообщение. |

|  |  |  |
| --- | --- | --- |
|  | ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  | Computer\_science |

|  |  |  |
| --- | --- | --- |
| **1. Теоретический материал** | | |
| Определим расстояние между символами кода. Пусть каждый символ кодируется последовательностью из *N* битов . Расстояние определим следующей формулой [1]:  где и 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 ≥ 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 ≥ 2n + 1,* позволяет найти количество ошибок, которые можно исправить. То есть, чтобы исправлять ошибок, необходимо учитывать расстояние между любыми символами, которое должно быть не меньше . | | |
| **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:*** | |
|  |  |
| 2. | ***Задача:*** | |
|  | Сколько ошибок можно исправить при использовании кодов из предыдущего примера? |
| ***Решение:*** | |
|  | Найдём минимальное расстояние между кодами.   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | A | 0 | 0 | 0 | 0 | 0 |  | B | 1 | 1 | 1 | 0 | 0 |  | C | 0 | 0 | 1 | 1 | 1 | | B | 1 | 1 | 1 | 0 | 0 |  | C | 0 | 0 | 1 | 1 | 1 |  | A | 0 | 0 | 0 | 0 | 0 | |  | | | | | |  |  | | | | | |  |  | | | | | |   →  Из этого условия найдем . Получим . 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:*** | |
|  | p(B,C) = p(1010100,0011110) = 3 |
| ***Ответ:*** | |
|  | 3 |

|  |  |  |
| --- | --- | --- |
| 2. | ***Задача:*** | |
|  | Верны ли следующие утверждения:  ,  ,  ? |
| ***Решение:*** | |
|  | p(A,B) =p(0101010,1010100)= 6  p(B,C) = p(1010100,0011110) = 3  p(A,C) = p(0101010,0011110) = 3  1.да  2.да  3.да |
| ***Ответ:*** | |
|  | 1.да 2.да 3.да |
| 3. | ***Задача:*** | |
|  | Сколько ошибок можно гарантированно исправить при использовании кодов из задачи 1? |
| ***Решение:*** | |
|  | Найдем d:  d = min(6,3,3)=3  d>=2n+1  3>=2n+1 |
| ***Ответ:*** | |
|  | Можно гарантированно исправить 1 ошибку |

|  |  |  |
| --- | --- | --- |
| **Тест 1** | | |
| **1.** | Сколько бит в 1 Кбайт? | |
|  | 1. 8192 бит 2. 75699 бит  3. 4589 бит 4. 34773 бит |
| ***Ответ:*** | |
|  | 1 |
| **2.** | Сколько бит нужно отвести на кодирование одной игральной карты стандартной колоды из 36 карт? | |
|  | 1. 4 2. 5 3. 6 4. 7 |
| ***Ответ:*** | |
|  | 3 |

|  |  |  |
| --- | --- | --- |
| **3.** | Сколько бит нужно отвести на кодирование двузначного десятеричного числа? | |
|  | 1. 6 2. 7 3. 8 4. 9 |
| ***Ответ:*** | |
|  | 4 |
| **4.** | Если не прогуливать занятия, то вероятность сдать сессии на «хорошо» и «отлично» равна 0,7. Найдите энтропию системы. | |
|  | 1. -0,05 2. 0,91 3. 0,13 4. 0,88 |
| ***Ответ:*** | |
|  | 4 |
| **5.** | Укажите ASCII код символа **G** | |
|  | 1. 67 2. 43 3. 6A 4. 47 |
| ***Ответ:*** | |
|  | 4 |
| **6.** | Бит четности служит для | |
|  | 1. исправления ошибок в данных  2. обнаружения ошибки в данных  3. шифрования данных  4. выравнивания данных |
| ***Ответ:*** | |
|  | 2 |
| **7.** | Если при пересылке сообщения в нём произошло ДВЕ ошибки, то бит четности | |
|  | 1. позволит исправить две ошибки  2. позволит их обнаружить  3. не позволит их обнаружить  4. позволит исправить только одну ошибку |
| ***Ответ:*** | |
|  | 3 |
| **8.** | В предположении, что в трех идущих подряд битах не может быть более одной ошибки, восстановите следующее сообщение: 001011101010100000001110 . | |
|  | 1. 01100001 2. 01101101  3. 11100001 4. 01000011 |
| ***Ответ:*** | |
|  | 1 |
| **9.** | Найдите, между какими кодами расстояние наибольшее | |
|  | 1. 11100010 и 00001111  2. 00010111 и 01110101  3. 01011010 и 10110101  4. 10101010 и 10101101 |
| ***Ответ:*** | |
|  | 3 |
| **10.** | 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 |
| ***Ответ:*** | |
|  | 4 |

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](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 . Значение вводится с клавиатуры. |
| ***Решение (код программы):*** | |
|  | x = input('Введите х') # 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(‘ Погосов Вячеслав Эдуардович ЭФБО-07-23’) |
| 2. | ***Задача:*** | |
|  | Введите с клавиатуры два числа и сложите их. Выведите результат на экран. |
| ***Решение (код программы):*** | |
|  | y = int(input())  x = int(input())  print(y + x) |
| 3. | ***Задача:*** | |
|  | Найти значение функции . Значение вводится с клавиатуры. |
| ***Решение (код программы):*** | |
|  | x = int(input())  print((x\*\*5)-(2\*x\*\*3)+1) |

|  |
| --- |
| **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) ) # функцияtypeвозвращает тип её аргумента |
| 2. | ***Задача:*** | |
|  | Определите истинность следующего выражения: |
| ***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 и , которые обратят выражение в значение True.  Выражение: . |
| ***Решение (код программы):*** | |
|  | for x in range(2):  for y in range(2):  statement = (x or y) and (not x or y) and not(x and y)  if bool(statement)==True:  print("x =", x, "y =", y) |
| 3. | ***Задача:*** | |
|  | Добавьте в пустой список четыре любых значения и выведете их на экран в обратном порядке, использую для этого индексы элементов. |
| ***Решение (код программы):*** | |
|  | list = [7,9,6,10]  list = list[::-1]  print(list) |

|  |  |  |
| --- | --- | --- |
| **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) #  b = m.sqrt( 16 ) #  c\_1 = m.e\*\*2 #  c\_2 = m.exp(2) #  d\_1 = m.log(8, 2) #  d\_2 = m.log2(8) #  e\_1 = m.ceil(3.14) # округление вверх (ответ 4)  e\_2 = m.ceil(2.7) # округление вверх (ответ 3) | | |
| **2. Пример** | | |
| 1. | ***Задача:*** | |
|  | Написать программу для решения квадратного уравнения, через дискриминант: . |
| ***Решение (код программы):*** | |
|  | 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) |
| 2. | ***Задача:*** | |
|  | Напишите программу для вычисления , где вводит пользователь с клавиатуры. |
| ***Решение (код программы):*** | |
|  | import math as m  x = float( input("Введите x: ") )  print( m.log2(7\*x)\*m.cos(x/3) ) |

|  |  |  |
| --- | --- | --- |
| **3. Задания** | | |
| 1. | ***Задача:*** | |
|  | Запрограммируйте формулы Хартли. Количество состояний вводится с клавиатуры. |
| ***Решение (код программы):*** | |
|  | *import* math *as* m K = int(input(“”)) H = m.ceil(m.log2(K)) print(H) |
| 2. | ***Задача:*** | |
|  | Запрограммируйте вычисление энтропии по Шеннону для систем из двух состояний. Вероятности вводятся с клавиатуры. |
| ***Решение (код программы):*** | |
|  | *import* math *as* m P1, P2 = float(input("")),float(input("")) H = -(P1\*m.log2(P1)+P2\*m.log2(P2)) print(H) |
| 3. | ***Задача:*** | |
|  | Напишите программу для вычисления , где вводит пользователь с клавиатуры. |
| ***Решение (код программы):*** | |
|  | *import* math *as* m x = float(input('')) y = (m.tan(m.cos(x) \* m.sin(2\*x)/(x\*m.exp(x))))\*\*m.log(x, 7) print(y) |
| 4. | ***Задача:*** | |
|  | Напишите программу для добавления бита четности к байту. Байт можете записать в виде списка (list) нулей и единиц. |
| ***Решение (код программы):*** | |
|  | b = [1,0,1,1,0,1,0,1] summa = sum(b) r = summa % 2 b.append(r) print(b) |