

Системы счисления

1. Чем KiB отличается от kB

1. Ничем не отличаются
2. В KiB 1024 бита, а в kB - 1024 байта
3. В KiB 1024 байт, а в kB - 1000 байт ☒
4. В KiB 1000 бита, а в kB - 1024 бит

2. Вычислите $0.1101_2 + 0.341_5$ и запишите в $_{10}$

3. $108_{10} \rightarrow _2$

4. В чем преимущество нега-позиционных и симметричных систем счисления?

1. Нет необходимости вводить дополнительный код для представления отрицательных чисел ☒
2. Занимают меньше места
3. Не нужно специального бита для обозначения отрицательных чисел ☒
4. Обладают более простыми арифметическими операциями

Кодирование данных

1. Для каких задач обработка данных с игнорированием ошибок будет приемлема?

1. Передача кодов ядерных установок
2. Составление списка студентов по профильной математике
3. Стриминг видео ☒
4. Установка программного обеспечения
5. Вывод статистики изменения стоимости акций ☒

2. Дано сообщение в коде Хэмминга 0100011. Каким было изначальное сообщение? Какой классический код использован?

Регулярные выражения и python

1. Какое из представленных ниже регулярных выражений является валидным выражением для поиска повторяющихся слов:

1. $(b\backslash w+\backslash b)(\backslash b)^+$ ☒
2. $(b[\backslash w+]\backslash b)^+$
3. $(b\backslash w+\backslash b)(\backslash 2\backslash b)^+$
4. $(\backslash w+ | \backslash w+)^*$

2. Какое из данных регулярных выражений корректно $(?=\backslash w+)$ или $(?<=\backslash w+)?$

1. Оба некорректны

2. Первое ☒
3. Второе
4. Первое и второе
3. Как в регулярном выражении обратиться к фрагменту текста, найденного в первой группе?
1. \1 ☒
2. \$1
3. ?1
4. ^1
4. Как в регулярном выражении повторить выражение из первой группы?
1. \1
2. \$1
3. ?1 ☒
4. ^1
5. Чем отличается re.findall() от re.finditer()?
1. Ничем не отличаются
2. finditer позволяет работать с группами ☒
3. findall возвращает список ☒
4. findall позволяет работать с группами
6. Что может принимать re.sub() в качестве второго аргумента?
1. Только строку
2. Любой элемент: строки, числа и тд
3. Строку или функцию ☒
4. Скомпилированное регулярное выражение
7. Зачем нужен re.compile()?

Языки разметки

1. Соответствует ли этот XML правилам (возможны несколько ответов)?
1. Да
2. Нет, в XML обязательно должна быть указана декларация ☒
3. Нет, текст не должен содержать символ *

```
<sequence name="start">
  <action cmd="triggeron">
    btn*
  </action>
  <action cmd="triggeron">
    msg_generic
  </action>
</sequence>
```

1. Приведенный ниже код относится к сериализации или десериализации данных? (сериализация)

1. Сериализация
2. Десериализация
3. Ни к тому, ни к другому
4. Не известно из условия

```
json.dump({'key': 'value'})
```

1. Что найдет регулярное выражение "`^<.+>`" в строке "`<div>simple div</div>`" (ответ `<div>simple div</div>`)

2. Что найдет регулярное выражение "`^<.+?>`" в строке "`<div>simple div</div>`" (ответ `<div>`)

Python

1. Что выведет программа (42)

1. True
2. False
3. None
4. 42

```
print(None or 42)
```

2. Что выведет программа (':3')

1. True
2. False
3. None
4. ':3'

```
print(':3' or 42)
```

3. Что выведет программа (None)

1. True
2. False
3. None
4. 42

```
print(None and 42)
```

4. Что выведет программа (42)

1. True
2. False
3. None
4. '3'

```
print(':3' and 42)
```

5. Что выведет программа (**None**)

```
def foo():  
    'hello, world!'  
  
print(foo())
```

6. Что означает ошибка "**IndentationError: unexpected indent**" в файле

```
def foo():  
    print('прошу отчислить по собственному желанию')  
    print('hello, world!')
```

7. Напишите сумму чисел, которые выведет программа ($2 + 2 + 2 + 3 = 9$)

```
for i in range(3):  
    if i % 12 == 11:  
        print(1)  
        break  
    else:  
        print(2)  
else:  
    print(3)
```