

11-класс, {1-уровень сложности}

1. Файл созданный в MS WORD имеет расширение:

- A) .XLSX
- B) .PPTX
- C) .DOCS
- D) .DOCX
- E) нет правильного ответа

2. Сервер — это:

- A) персональный компьютер, подключенный к сети, через который пользователь получает доступ к ее ресурсам
- B) компьютер, подключенный к сети и обеспечивающий ее пользователей определенными услугами
- C) два или более абонентов вычислительной сети, соединенных каналом связи.
- D) персональный компьютер
- E) нет правильного ответа

3. Наиболее эффективными средствами защиты от компьютерных вирусов являются:

- A) антивирусные программы
- B) операционная система
- C) аппаратные средства
- D) организационные мероприятия
- E) базовые средства

4. FAT32, Ext2, NTFS - это...

- A) Расширения файлов
- B) Названия различных операционных систем
- C) Виды кодировки файлов
- D) Названия различных файловых систем
- E) нет правильного ответа

5. Какой фирмой выпускается операционная система MacOS?

- A) Acer
- B) IBM
- C) HP
- D) Apple
- E) нет правильного ответа

6. Jpg, gif, png, tiff - это...

- A) Расширения видео файлов
- B) Названия различных файловых систем
- C) Расширения текстовых файлов
- D) Расширения графических файлов
- E) Расширения аудио файлов

7. Какая единица информации является наибольшей?

- A) килобайт
- B) гигабайт
- C) петабайт
- D) эксабайт
- E) терабайт

8. Компьютер хранит все данные в виде чисел:

- A) 0 и 1
- B) 0 и 2
- C) 1 и 2
- D) 0 и 3
- E) 1 и 3

9. На панели задач находятся ...

- A) кнопки свернутых программ
- B) только ярлыки
- C) кнопка "Пуск"
- D) кнопка "Пуск" и значки свернутых и работающих программ
- E) нет правильного ответа

10. Количество значащих нулей в двоичной записи числа 289 равно:

- A) 4
- B) 5
- C) 6
- D) 7
- E) 8

11-класс, {2-уровень сложности}

11.

```
a, b, c = 7, 5.3, "Name"
```

```
print (a)
```

```
print (b)
```

```
print (c)
```

Согласно коду указанному выше чему равно значение переменной a?

A) 7

B) 5

C) 5.3

D) Name

E) Error

12.

```
x = 6
```

```
y = 14
```

```
x, y = y, x
```

```
print("x =", x)
```

```
print("y =", y)
```

Какой результат выдает программа указанная выше?

A) Выводить начальные значения x и y

B) Приравнивает значения x и y к 6

C) Меняет значение x и y, и выводит их новое значение

D) Выводить сумму значений x и y

E) Приравнивает значения x и y к 14

13.

```
a = int(input("Enter first coefficient: "))
```

```
b = int(input("Enter second coefficient: "))
```

```
c = int(input("Enter third coefficient: "))
```

```
d = (b2) - (4ac)
```

```
sol1 = (-b-d0.5)/(2a)
```

```
sol2 = (-b+d0.5)/(2a)
```

```
print(sol1,sol2)
```

Что вычисляет программа указанная выше?

A) Вычисляет корень линейного уравнения

B) Вычисляет корни квадратного уравнения

C) Вычисляет корни уравнения с двумя неизвестными

D) Вычисляет корни уравнения с тремя неизвестными

E) Вычисляет корни уравнения с четырьмя неизвестными

```
14. for v in "strike":  
    if v == "i":  
        continue  
    print(v)  
print("End")
```

Какой результат выводит вышеуказанная программа?

A)s
End

B)s
t
End

C)s
t
r
End

D)s
t
r
i
End

E) s
t
r
k
e
End

```
15. numbers = [7, 9, 2, 6, 3, 1, 15, 18, 23]  
sum = 0  
for val in numbers:  
    sum = sum+val  
print(sum)
```

Какой результат выводит вышеуказанная программа?

- A)80
- B)82
- C) 84
- D)86
- E)88

16. Допустим список **lst** имеет размер **N** который нам не известен. Как можно вывести последний элемент списка?

- A) `print(lst[N])`
- B) `print(lst[-1])`
- C) `print(lst.last())`
- D) `print(lst[0])`
- E) `print(len(lst))`

17. Этот код должен посчитать четные и нечетные цифры числа **a**. Но в нем есть ошибка. Найдите ее

```
a = input()  
a = int(a)  
even = 0  
odd = 0  
while a > 0:  
    if a % 2 == 0:  
        even += 1  
    else:  
        odd += 1  
    a = a % 10  
print("Even: %d, odd: %d" % (even, odd))
```

- A) переменные `even` и `odd` должны начинаться с 1
- B) вместо `a%2` должно быть `a//2`
- C) вместо `a%10` должно быть `a//10`
- D) вместо знака `>` должно быть `<`
- E) в коде нет ошибки

18. Чему будут равны **x** и **y** после выполнения этого кода?

```
x = 5  
y = 6  
x, y = y, x  
print("x:", x, "y:", y)
```

- A) 6, 5
- B) 5, 6
- C) 5, 5
- D) 6, 6
- E) 0, 0

19. Что выдаст этот код?

```
def sum_digits(num):  
    digits = [int(d) for d in str(num)]  
    return sum(digits)  
print(sum_digits(5245))
```

- A) 15
- B) 13
- C) 18
- D) 0
- E) 16

20. Программа, которая выводит чётные числа из заданного списка и останавливается, если встречается число 237

```
A) numbers = [  
    386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345,  
    399, 162, 758, 219, 918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217,  
]
```

```
for x in numbers:  
    if x == 237:  
        break  
    elif x % 2 == 0:  
        print(x)
```

```
B) numbers = [  
    386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345,  
    399, 162, 758, 219, 918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217,  
]
```

```
for x in numbers:  
    if x == 237:  
        break  
    elif x // 2 == 0:  
        print(x)
```

```
C) numbers = [  
    386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345,  
    399, 162, 758, 219, 918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217,  
]  
for x in numbers:  
    if x == 237:  
        break  
    elif x % 2 != 0:  
        print(x)
```

```
D) numbers = [  
    386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345,  
    399, 162, 758, 219, 918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217,  
]  
for x in numbers:  
    if x == 257:  
        break  
    elif x % 2 == 0:  
        print(x)
```

```
E) numbers = [  
    386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345,  
    399, 162, 758, 219, 918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217,  
]  
for x in numbers:  
    if x == 237:  
        continue  
    elif x % 2 == 0:  
        print(x)
```

11-класс, {3-уровень сложности}

21.

```
a = 0b1010100100 #Двоичное значение  
b = 24000 #Десятичное значение  
c = 0o422367 #Восьмеричное значение  
d = 0x89cdef #Шестнадцатеричное значение  
print(a, b, c, d)
```

Какой результат выводит программа указанная выше?

- A) 676 24000 140533 9031151
- B) 676 24000 140535 9031151
- C) 676 24000 140537 9031151
- D) 676 24000 140535 9031152
- E) 676 24000 140535 9031153

22.

```
lower = int(input())  
upper = int(input())  
for num in range(lower, upper + 1):  
    if num > 1:  
        for i in range(2, num):  
            if (num % i) == 0:  
                break  
        else:  
            print(num)
```

Что определяет вышеуказанная программа?

- A) Все четные числа в интервале между значениями lower и upper
- B) Все нечетные числа в интервале между значениями lower и upper
- C) Все простые числа в интервале между значениями lower и upper
- D) Все числа кратные 3-м в интервале между значениями lower и upper
- E) Все составные числа в интервале между значениями lower и upper

23. Какая из нижеуказанных программ из заранее заданного списка выводит в новом списке элементы и количество их повторений в предыдущем списке?

```
A) import set  
my_list = [10,10,10,10,20,20,20,20,40,40,50,50,30]  
ctr = collections.Counter(my_list)  
print("Frequency of the elements in the List : ",ctr)  
B) import math  
my_list = [10,10,10,10,20,20,20,20,40,40,50,50,30]  
ctr = collections.Counter(my_list)  
print("Frequency of the elements in the List : ",ctr)
```

```

C) import collections
my_list = [10,10,10,10,20,20,20,20,40,40,50,50,30]
ctr = collections.Counter(my_list)
print("Frequency of the elements in the List : ",ctr)
D) import set
my_list = [10,10,10,10,20,20,20,20,40,40,50,50,30]
str = collections.Counter(my_list)
print("Frequency of the elements in the List : ",ctr)
E) import math
my_list = [10,10,10,10,20,20,20,20,40,40,50,50,30]
add = collections.Counter(my_list)
print("Frequency of the elements in the List : ",ctr)

```

24. Какая из нижеуказанных программ определяет является ли число числом Армстронга или не является (Число Армстронга — натуральное число, которое в данной системе счисления равно сумме своих цифр, возведенных в степень, равную количеству его цифр. Например, десятичное число 153 — число Армстронга, потому что $1^3 + 5^3 + 3^3 = 153$)?

```

A) num = int(input("Enter a number: "))
sum = 0
temp = num
while temp > 0:
    digit = temp % 10
    sum += digit 3
    temp /= 10
if num == sum:
    print(num,"is an Armstrong number")
else:
    print(num,"is not an Armstrong number")

```

```

B) num = int(input("Enter a number: "))
sum = 0
temp = num
while temp > 0:
    digit = temp / 10
    sum += digit 3
    temp //= 10
if num == sum:
    print(num,"is an Armstrong number")
else:

```

```
print(num,"is not an Armstrong number")
```

```
C) num = int(input("Enter a number: "))
```

```
sum = 0
```

```
temp = num
```

```
while temp > 0:
```

```
    digit = temp % 10
```

```
    sum += digit ** 3
```

```
    temp //= 10
```

```
if num == sum:
```

```
    print(num,"is an Armstrong number")
```

```
else:
```

```
    print(num,"is not an Armstrong number")
```

```
D) num = int(input("Enter a number: "))
```

```
sum = 0
```

```
temp = num
```

```
while temp > 0:
```

```
    digit = temp % 10
```

```
    sum += digit ** 3
```

```
    temp /= 10
```

```
if num == sum:
```

```
    print(num,"is an Armstrong number")
```

```
else:
```

```
    print(num,"is not an Armstrong number")
```

```
E) num = int(input("Enter a number: "))
```

```
sum = 0
```

```
temp = num
```

```
while temp > 0:
```

```
    digit = temp % 10
```

```
    sum += digit ** 3
```

```
    temp //= 10
```

```
if num == sum:
```

```
    print(num,"is an Armstrong number")
```

```
else:
```

```
    print(num,"is not an Armstrong number")
```

25. Какая из нижеуказанных программ выводит все числа Армстронг в интервале между lower и upper (Число Армстронга — натуральное число, которое в данной системе счисления равно сумме своих цифр, возведенных в степень, равную количеству его цифр. Например, десятичное число 153 — число Армстронга, потому что $1^3 + 5^3 + 3^3 = 153$)?

```
A) lower = int(input())
upper = int(input())
for num in range(lower, upper + 1):
    order = len(str(num))
    sum = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit ** order
        temp //= 10
    if num == sum:
        print(num)
```

```
B) lower = int(input())
upper = int(input())
for num in range(lower, upper + 1):
    order = len(str(num))
    sum = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit * order
        temp /= 10
    if num == sum:
        print(num)
```

```
C) lower = int(input())
upper = int(input())
for num in range(lower, upper + 1):
    order = len(str(num))
    sum = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit * order
        temp //= 10
    if num == sum:
        print(num)
```

```

D) lower = int(input())
upper = int(input())
for num in range(lower, upper + 1):
    order = len(str(num))
    sum = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum = digit * order
        temp //= 10
    if num == sum:
        print(num)

```

```

E) lower = int(input())
upper = int(input())
for num in range(lower, upper + 1):
    order = len(str(num))
    sum = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit * order
        temp //= 10
    if num == sum:
        print(num)

```

26. Что выдаст этот код? Палиндром — это слово или фраза, которые одинаково читаются слева направо и справа налево

```

def is_palindrome(string):
    return string == string[::-1]
print(is_palindrome('abba'))

```

- A) False
- B) abba
- C) True
- D) a
- E) Yes

27. Программа которая отображает число секунд в формате
дни:часы:минуты:секунды

```
A)def convert():
    days = seconds // (24 * 3600)
    seconds %= 24 * 3600
    hours = seconds // 3600
    seconds %= 3600
    minutes = seconds // 60
    seconds %= 60
    print(f'{days}:{hours}:{minutes}:{seconds}')
```

```
B)def convert(seconds):
    days = seconds // (24 * 3600)
    seconds %= 24 * 3600
    hours = seconds // 3600
    seconds %= 3600
    minutes = seconds // 60
    seconds //= 60
    print(f'{days}:{hours}:{minutes}:{seconds}')
```

```
C)def convert(seconds):
    days = seconds // (24 * 3600)
    seconds %= 24 * 3600
    hours = seconds // 3600
    seconds %= 3600
    minutes = seconds // 60
    seconds %= 60
    print(f'{days}:{hours}:{minutes}:{seconds}')
```

```
D)def convert(seconds):
    days = seconds // (24 * 3600)
    seconds %= 24 * 3600
    hours = seconds // 3600
    seconds %= 60
    minutes = seconds // 60
    seconds %= 60
    print(f'{days}:{hours}:{minutes}:{seconds}')
```

```
E)def convert(seconds):
    days = seconds // (24 * 3600)
    seconds %= 24 * 3600
    hours = seconds // 3600
    seconds %= 3600
    minutes = seconds // 60
    seconds %= 60
    print(f'{days}:{hours}:{minutes}:{seconds}')
```

28. Каков будет ответ данного кода

```
def solve(n):
    n1 = n
    n2 = int(str(n) 2)
    n3 = int(str(n) 3)
    print(n1 + n2 + n3)
solve(5)
```

- A)615
- B)30
- C)15
- D)555
- E)523

29. Если даны два списка данная программа находит их совпадающие элементы. Код ниже подходит для списков, содержащих неповторяющиеся значения в самих себе. Иначе в результирующем списке могут появиться одинаковые элементы. Если в самом списке могут встречаться одинаковые элементы, чтобы они не повторялись что следует добавить вместо ... в результирующем списке, с.

```
a = [5, [1, 2], 2, 'r', 4, 'ee', 'ee']
```

```
b = [4, 'we', 'ee', 3, [1, 2]]
```

```
c = []
```

```
for i in a:
```

```
    ...
```

```
        for j in b:
```

```
            if i == j:
```

```
                c.append(i)
```

```
                break
```

```
print(c)
```

A) if i in c:

```
    break
```

B) if i in c:

```
    continue
```

C) for i in c:

```
    continue
```

D) for i in c:

```
    break
```

E) if i in c:

```
    c.append(i)
```

30. Вывести степени натуральных чисел, не превосходящие данного числа n. Пользователь задает показатель степени и число n. (hard)

Алгоритм решения задачи:

1. Ввести показатель степени и присвоить его переменной p.
2. Ввести число n.
3. Пока натуральное число i возведенное в степень p меньше или равно n, то
 - a. выводить на экран i в степени p и
 - b. увеличивать i на 1 (т. е. переходить к следующему натуральному числу).

Какая программа соответствует данному алгоритму?

```
A)p = int(input("Показатель степени: "))
n = int(input("Предел: "))
i = 1
while i p <= n:
    print(i p, end=' ')
    i += 1
print("\nПоследнее число, возведенное в степень:", i - 1)
```

```
B)p = int(input("Показатель степени: "))
n = int(input("Предел: "))
i = 1
while i p <= n:
    print(i p, end=' ')
    i += 1
print("\nПоследнее число, возведенное в степень:", i - 1)
```

```
C)p = int(input("Показатель степени: "))
n = int(input("Предел: "))
i = 1
while i p <= n:
    print(i p, end=' ')
    i += 1
print("\nПоследнее число, возведенное в степень:", i - 1)
```

```
D)p = int(input("Показатель степени: "))
n = int(input("Предел: "))
i = 1
while i p < n:
    print(i p, end=' ')
    i += 1
print("\nПоследнее число, возведенное в степень:", i - 1)
```

```
E)p = int(input("Показатель степени: "))
n = int(input("Предел: "))
i = 1
while i p <= n:
    print(i p, end=' ')
    i += 1
print("\nПоследнее число, возведенное в степень:", i - 1)
```