1. def sum\_a(n):

sum = 0

while n > 0:

b = n % 10

sum += b

n = n // 10

return sum

number = 1234567

print('сумма цифр', number, '=', sum\_a(number))

----------------------------------------------------------------------------------------------------

2. def a\_b(n):

a\_b = 0

while n > 0:

b = n % 10

a\_b= (a\_b \* 10) + b

n = n // 10

return a\_b

number = 123454

print("Перевернутое число", number, " =", a\_b(number))

----------------------------------------------------------------------------------------------------

3.def a\_q(c, d):

b = []

for number in range(c, d + 1):

if number > 1:

is\_prime = True

for i in range(2, number):

if number % i == 0:

is\_prime = False

break

if is\_prime:

b.append(number)

return b

b = a\_q(2, 100)

print("В диапазоне от 2 до 100 найдено", len(b), "простых чисел:")

print(b)

----------------------------------------------------------------------------------------------------

4. def a\_w(qwe):

return qwe[::-1]

a = [1, 2, 3, 4, 5]

print("Исходный список:", a)

a\_w = a\_w(a)

print("Обратный список:", a\_w)

----------------------------------------------------------------------------------------------------

5. a = [1, 2, 3, 4, 5, 6, 7]

b= [x \*\* 2 for x in a]

print(b)

----------------------------------------------------------------------------------------------------

6. def a\_a(lst):

lst.sort(key=lambda x: abs(x), reverse=True)

return lst

print(a\_a([1, 5, 77, 101, 10091]))

----------------------------------------------------------------------------------------------------

7.a= ['Apple', 'Grape', 'Peach', 'Banan', 'Orange']

b = a[0:3]

print(b)

----------------------------------------------------------------------------------------------------

8. def a\_t(var1, var2):

temp = var1

var1 = var2

var2 = temp

return (var1, var2)

a = 1

b = 2

a, b = a\_t(a,b)

print("a =",a)

print("b =",b)

----------------------------------------------------------------------------------------------------

9. def is\_palindrome(sentence):  
 sentence\_without\_spaces = sentence.replace(" ", "")  
 return sentence\_without\_spaces == sentence\_without\_spaces[::-1]  
  
print(is\_palindrome("АРГЕНТИНА МАНИТ НЕГРА"))   
print(is\_palindrome("Экзаменационные Вопросы"))

----------------------------------------------------------------------------------------------------

10. week\_temp = (26, 29, 34, 32, 28, 26, 23)  
num\_days = len(week\_temp)  
avg\_temp = sum(week\_temp) / num\_days  
print("Количество дней в неделе:", num\_days)  
print("Средняя температура:", avg\_temp)

----------------------------------------------------------------------------------------------------

11. digits = (1, 2, 3, 4, 5)  
digits\_sum = sum(digits)  
print("Сумма цифр:", digits\_sum)

----------------------------------------------------------------------------------------------------

12. def sum\_numbers(n):  
 return sum(range(1, n+1))  
N = int(input('Введите число:'))  
sum\_N = sum\_numbers(N)  
print(f"Сумма чисел от 1 до {N}:", sum\_N)

----------------------------------------------------------------------------------------------------

13. heights = [170, 165, 175, 180, 160, 168, 172, 165, 173, 176, 169, 171, 170, 175, 162, 168, 169, 166, 174, 163, 167, 172]  
r = 170  
count = sum(1 for h in heights if h <= r)  
print(count)

----------------------------------------------------------------------------------------------------

14. def to\_set(data):  
 s = set(data)  
 return s, len(s)  
  
data = [1, 2, 3, 4, 5, 5]  
result\_set, result\_len = to\_set(data)  
print(result\_set)  
print(result\_len)

----------------------------------------------------------------------------------------------------

15. def count\_vowels(string):  
 vowels = set('aeiouAEIOU')  
 count = sum(1 for c in string if c in vowels)  
 return count  
  
string = 'Hello, World!'  
print(count\_vowels(string))

----------------------------------------------------------------------------------------------------

16. def common\_chars(str1, str2):  
 set1 = set(str1)  
 set2 = set(str2)  
 common = set1 & set2  
 return common  
  
str1 = 'hello'  
str2 = 'world'  
print(common\_chars(str1, str2))

----------------------------------------------------------------------------------------------------

17. def m(a):  
 p = 1  
 for r in a:  
 for i in r:  
 p \*= i  
 return p  
  
a = [[1, 2, 3, 4], [8, 9, 10, 12]]  
print(m(a))

----------------------------------------------------------------------------------------------------

18. a = [[1, 2, 3, 4], [8, 9, 10, 12]]  
def sum(a):  
 t\_sum = 0  
 for r in a:  
 for e in r:  
 t\_sum += e  
 return t\_sum  
print(sum(a))

----------------------------------------------------------------------------------------------------

19. keys = ['red', 'green', 'blue']  
values = ['Apple', 'Gras', 'Sky']  
dictionary = {keys[i]: values[i] for i in range(len(keys))}  
print(dictionary)

----------------------------------------------------------------------------------------------------

20. class RomanToInt:  
 def \_\_init\_\_(self):  
 self.r = {"I": 1, "V": 5, "X": 10, "L": 50, "C": 100, "D": 500, "M": 1000}  
  
 def r\_to\_int(self, s):  
 r = 0  
 p = 0  
 for c in s:  
 c = self.r[c]  
 r += c  
 if c > p:  
 r -= 2 \* p  
 prev = c  
 return r  
rc = RomanToInt()  
rn = (input('Введите римскую цифру: '))  
i\_n = rc.r\_to\_int(rn)  
print("Римская цифра {} эквивалентна целому числу {}".format(rn, i\_n))

----------------------------------------------------------------------------------------------------

21. class IntToRoman:  
 def \_\_init\_\_(self, num):  
 self.num = num  
 self.r\_digits = [("M", 1000), ("CM", 900), ("D", 500), ("CD", 400),  
 ("C", 100), ("XC", 90), ("L", 50), ("XL", 40),  
 ("X", 10), ("IX", 9), ("V", 5), ("IV", 4), ("I", 1)]  
  
 def to\_r(self):  
 if not isinstance(self.num, int) or self.num <= 0:  
 return "Ошибка: введите натуральное число"  
  
 r\_num = ""  
 for digit, val in self.r\_digits:  
 while self.num >= val:  
 r\_num += digit  
 self.num -= val  
  
 return r\_num  
c = int(input('Введите число: '))  
cr = IntToRoman(c)  
print(cr.to\_r())

----------------------------------------------------------------------------------------------------

22. class Circle:  
 def \_\_init\_\_(self, radius):  
 self.r = radius  
  
 def s(self):  
 return 3.14 \* (self.r \*\* 2)  
  
 def p(self):  
 return 2 \* 3.14 \* self.r  
c = float(input('Введите радиус: '))  
cl = Circle(c)  
print("Площадь круга:", cl.s())  
print("Периметр круга:", cl.p())

----------------------------------------------------------------------------------------------------

23. class Rectangle:  
 def \_\_init\_\_(self, length, width):  
 self.l = length  
 self.w = width  
  
 def calculate\_area(self):  
 s = self.l \* self.w  
 return s  
  
length = float(input("Введите длину прямоугольника: "))  
width = float(input("Введите ширину прямоугольника: "))  
  
  
r = Rectangle(length, width)  
s = r.calculate\_area()  
print(f"Площадь прямоугольника с длиной {length} и шириной {width} равна {s}")

----------------------------------------------------------------------------------------------------

24. class StringManipulator:  
 def \_\_init\_\_(self):  
 self.n\_str = ""  
  
 def get\_string(self):  
 self.n\_str = input("Введите строку: ")  
  
 def print\_string(self):  
 print(self.n\_str.upper())  
  
m\_str = StringManipulator()  
m\_str.get\_string()  
m\_str.print\_string()

----------------------------------------------------------------------------------------------------

25. class KgToPounds:  
 def \_\_init\_\_(self, kg):  
 self.kg = kg  
  
 def to\_pounds(self):  
 pounds = self.kg \* 2.20462  
 return pounds  
p = float(input('Введите массу: '))  
w = KgToPounds(p)  
pounds = w.to\_pounds()  
print(f"{p} килограмм = {pounds} фунтов")

----------------------------------------------------------------------------------------------------

26. class Voda:  
 def \_\_init\_\_(self, dobavka):  
 self.dobavka = dobavka  
  
 def show\_my\_drink(self):  
 if self.dobavka:  
 print(f"Газировка и {self.dobavka}")  
 else:  
 print("Обычная газировка")  
# Девочки в "" добавляем любой фрукт или овощ, если я правильно поняла задание, то код верный  
  
v = Voda(None)  
v.show\_my\_drink()