

### Задание 1.

Исходные данные:

Найти сумму и произведение цифр трехзначного числа, которое вводит пользователь.

Решение:

Текстовый алгоритм.

1. Мы просим пользователя ввести с клавиатуры трёхзначное число.
2. Полученное число мы делим на сто без остатка, чтобы далее было с ним комфортно работать.
3. Далее мы из полученного числа на шаге два, создаём сумму из трёх одинаковых чисел.
4. Далее мы из полученного числа на шаге два получаем произведение, сначала умножая число на десять, а затем ещё раз на одиннадцать.
5. Получившиеся результаты на шаге три и четыре, мы выводим на экран, чтобы пользователь их видел.

Код из Python иллюстрирующий данный текстовой алгоритм.

```
a = int(input("Введите целое трёхзначное число: "))
```

```
b = a // 100
```

```
c = b + b + b
```

```
d = b * 10 * 11
```

```
print(c, d)
```

### Задание 2.

Исходные данные:

Выполнить логические побитовые операции "И", "ИЛИ" и др. над числами 5 и 6.

Выполнить над числом 5 побитовый сдвиг вправо и влево на два знака.

Решение:

Текстовый алгоритм.

1. Мы просим пользователя ввести с клавиатуры целое число.
2. Мы просим снова ввести с клавиатуры пользователя целое число.
3. Далее мы выполняем операцию, побитовое (И).
4. Выводим на экран получившейся результат.
5. Мы снова просим ввести пользователя с клавиатуры целое число.
6. Мы снова просим пользователя ввести с клавиатуры целое число.
7. Далее мы выполняем операцию побитовое (ИЛИ).
8. Выводим получившейся результат на экран.
9. Мы снова просим пользователя ввести с клавиатуры целое число.
10. Далее мы выполняем операцию побитового сдвига в лево.
11. Выводим получившейся результат на экран.
12. Мы снова просим пользователя ввести с клавиатуры целое число.
13. Далее мы выполняем операцию побитового сдвига в право.
14. Выводим получившейся результат на экран.

Вот код на Python иллюстрирующий данный текстовой алгоритм.

```
a = int(input("Введите целое число: "))
```

```
b = int(input("Введите целое число: "))
```

```
c = a & b
```

```
print(c)
```

```
d = int(input("Введите целое число: "))
```

```
e = int(input("Введите целое число: "))
```

```
f = d | e
```

```
print(f)
```

```
g = int(input("Введите целое число: "))
```

```
k = g << 2
```

```
print(k)
```

```
h = int(input("Введите целое число: "))
```

```
l = h>>2  
print(l)
```

### Задание 3.

Исходные данные:

По введенным пользователем координатам двух точек вывести уравнение прямой вида  $y = kx + b$ , проходящей через эти точки.

Решение:

Текстовый алгоритм.

1. Мы просим у пользователя ввести с клавиатуры целое число.
2. Далее мы работаем с формулой, которая нам была предоставлена изначально.
3. Согласно этой формуле, так как у нас дана только одна прямая, то коэффициент  $k$  равен

1.

4. Далее следуя данной формуле записываем свободный член,  $b$  равным 0.
5. далее мы подставляем все данные в формулу и получаем результат.
6. Полученный результат мы выводим на экран.

Вот код на Python иллюстрирующий текстовый алгоритм.

```
a = 7  
y = 7  
def coordinats(a, y):  
    try:  
        a = int(input("Введите целое число: "))  
        k = 1  
        b = 0  
    except zerodivisionerror:  
        return  
    y = k * a + b  
    return y  
  
print(coordinats(a,y))
```

### Задание 4.

Исходные данные:

Написать программу, которая генерирует в указанных пользователем границах

- случайное целое число,
- случайное вещественное число,
- случайный символ.

Для каждого из трех случаев пользователь задает свои границы диапазона. Например, если надо получить случайный символ от 'a' до 'f', то вводятся эти символы. Программа должна вывести на экран любой символ алфавита от 'a' до 'f' включительно.

Решение:

- случайное целое число,

Текстовый алгоритм.

1. Мы просим пользователя ввести с клавиатуры целое число.
2. Мы снова просим пользователя ввести с клавиатуры целое число.
3. Далее мы используем библиотеку генерации псевдочисел.
4. Из данной библиотеки мы используем подходящий нам метод, для решения данной задачи.
5. Мы сразу передаём значения полученные от пользователя в данный метод и одновременно сразу же выводим получившейся результат работы данного метода на экран.

Вот код на Python иллюстрирующий данный текстовый алгоритм.

```
a = int(input("Введите целое число: "))  
b = int(input("Введите целое число: "))
```

```
import random
from random import randrange
print(randrange(a, b))
```

случайное вещественное число.

Текстовый алгоритм.

1. Мы просим пользователя ввести с клавиатуры вещественное число.
2. Мы снова просим пользователя ввести с клавиатуры вещественное число.
3. Далее мы используем библиотеку генерации псевдочисел.
4. Из данной библиотеки мы используем подходящий нам метод, для решения данной задачи.
5. Мы сразу передаём значения полученные от пользователя в данный метод и одновременно сразу же выводим получившейся результат работы данного метода на экран.

Вот код на Python иллюстрирующий данный текстовый алгоритм.

```
a = float(input("Введите вещественное число: "))
b = float(input("Введите вещественное число: "))
import random
from random import uniform
print(uniform(a, b))
```

- 
- случайный символ.

Текстовый алгоритм.

1. Для начала мы импортируем библиотеку для псевдочисел и псевдосимволов.
2. Мы просим пользователя ввести с клавиатуры набор букв.
3. Далее мы этот набор букв передаём в функцию.
4. Далее мы сначала производим внутреннюю печать результата, но не выводим этот результат на экран.
5. Далее генерируем необходимый результат, на основе данной функции, который мы выводим на экран.

Вот код на Python иллюстрирующий данный текстовый алгоритм.

```
import random
a = input("Введите буквы: ")
def generate_random_string(a):
    letters = 'abcdef'
    rand_string = ''.join(random.choice(letters) for i in range(a))
    print("Random string of a", a, "is:", rand_string, a)
```

```
generate_random_string(1)
```

Задание 5.

Исходные данные:

Пользователь вводит две буквы. Определить, на каких местах алфавита они стоят, и сколько между ними находится букв.

Решение:

Текстовый алгоритм.

1. Мы просим пользователя ввести с клавиатуры строчную букву латинского алфавита.
2. Мы снова просим пользователя ввести строчную букву латинского алфавита.
3. Далее мы используем формулы для определения, какую по счёту строчную букву алфавита нам пользователь ввёл.
4. Далее мы используем формулу для определения, сколько будет символов между двумя буквами, которые нам ввёл пользователь..
5. Далее выводим получившейся результат на экран.

Вот код на Python иллюстрирующий данный текстовый алгоритм.

```
a = 'b'
b = 'd'
def away(a, b):
    try:
        a = input("Введите строчную букву латинского алфавита: ")
        b = input("Введите строчную букву латинского алфавита: ")
    except ValueError and ZeroDivisionError:
        return
    c = ord(a) - 97 + 1
    d = ord(b) - 97 + 1
    f = ord(b) - ord(a)
    return c, d, f

print(away(a, b))
```

Задание 6.

Исходные данные:

Пользователь вводит номер буквы в алфавите. Определить, какая это буква.

Решение:

Текстовый алгоритм.

1. Мы просим пользователя ввести с клавиатуры номер буквы.
2. Далее мы используем формулу для определения буквы алфавита, по номеру, который нам ввёл пользователь.
3. Далее выводим полученный результат на экран.

Вот код на Python иллюстрирующий данный текстовый алгоритм.

```
a = 2
def away(a):
    try:
        a = int(input("Введите номер буквы латинского алфавита: "))
    except ZeroDivisionError:
        return

    s = chr(a)
    return s

print(away(a))
```

Задание 7.

Исходные данные:

По длинам трех отрезков, введенных пользователем, определить возможность существования треугольника, составленного из этих отрезков. Если такой треугольник существует, то определить, является ли он разносторонним, равнобедренным или равносторонним.

Решение:

Текстовый алгоритм.

1. Мы просим ввести пользователя с клавиатуры число.
2. Мы снова просим ввести пользователя с клавиатуры число.
3. Мы снова просим пользователя ввести с клавиатуры число.
4. далее мы используем разветвлённый цикл для того, чтобы определить, какого рода получится фигура.
5. Далее мы выводим полученный результат на экран.

Вот код на Python иллюстрирующий данный текстовый алгоритм.

```
a = 6
b = 6
c = 6
def treug(a, b, c):
    try:
        a = int(input("Введите число: "))
        b = int(input("Введите число: "))
        c = int(input("Введите число: "))
    except zerodivisionerror:
        return
    if a+b>c:
        print('Равнобедренный')
    else:
        print('Равносторонний')

print(treug(a, b, c))
```

Задание 8.

Исходные данные:

Определить, является ли год, который ввел пользователем, високосным или не високосным.

Решение:

Текстовый алгоритм.

1. Мы просим пользователя ввести с клавиатуры число.
2. Далее мы используем формулу для обработки данного числа.
3. Далее мы выводим получившейся результат на экран.

Вот код на Python иллюстрирующий данный текстовый алгоритм.

```
a = 2021
def year(a):
    try:
        a = int(input("Введите год числом: "))
    except zerodivisionerror:
        return
    b = a // 4
    c = a // 400
    d = a // 100 != 0
    s = a // b and c // d
    return s

print(year(a))
```

Задание 9.

Исходные данные:

Вводятся три разных числа. Найти, какое из них является средним (больше одного, но меньше другого).

Решение:

Текстовый алгоритм.

1. Мы просим пользователя ввести с клавиатуры число.
2. Мы снова просим ввести с клавиатуры число.
3. Мы снова просим пользователя ввести с клавиатуры число.

4. Далее мы используем необходимые формулы для обработки полученных данных от пользователя.

5. Далее мы выводим полученный результат на экран.

Вот код на Python иллюстрирующий данный текстовый алгоритм.

```
a = 205
```

```
b = 175
```

```
c = 985
```

```
def sravn(a, b, c):
```

```
    try:
```

```
        a = int(input("Введите число: "))
```

```
        b = int(input("Введите число: "))
```

```
        c = int(input("Введите число: "))
```

```
    except zerodivisionerror:
```

```
        return
```

```
    d = a > b or a < b
```

```
    e = a > c or a < c
```

```
    f = b > c or b < c
```

```
    g = c > a or c < a
```

```
    h = c > b or c < b
```

```
    if a>b>c:
```

```
        print(a)
```

```
    if b>a>c:
```

```
        print(b)
```

```
    if c>b>a:
```

```
        print(c)
```

```
    return d, e, f, g, h
```

```
print(sravn(a, b, c))
```