

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ”
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №1

Специальность ПО-13

Выполнил:
М.А.Шумило
студент группы
ПО-13
Проверил:
А.Д.Кулик
13.02.2026

Брест 2026

Цель работы: закрепить базовые знания языка программирования Python при решении практических задач.

Задание 1.

9) Распределение чисел: количество одноцифрных чисел, двуцифрных, трехцифрных и т. д.

Выполнение:

Код программы:

```
while True:
    raw = input("Введите числа через пробел: ").split()
    nums = []
    ok = True
    for item in raw:
        try:
            nums.append(int(item))
        except ValueError:
            print(f"Ошибка: '{item}' не является целым числом. Попробуйте снова.
\n")
            ok = False
            break
    if ok:
        break

dist = {}
for n in nums:
    digits = len(str(abs(n)))
    if digits not in dist:
        dist[digits] = 0
    dist[digits] += 1

print("\nРаспределение по количеству цифр:")
for digits in sorted(dist):
    print(f"{digits}-значных: {dist[digits]}")
```

Спецификация ввода

Введите числа через пробел: <1-е число> <2-е число> <3-е число>

Пример

Введите числа через пробел: 11 203 4003

Спецификация вывода

Распределение по количеству цифр:
1-значных:<количество 1-значных>
2-значных:<количество 2-значных>
3-значных:<количество 3-значных>

Пример

Распределение по количеству цифр:

1-значных: 1

2-значных: 5

3-значных: 4

Рисунки с результатами работы программы

```
PS C:\Users\user\Desktop\СПП\spp_po13\reports\Shumilo\1\src> python 1.py
Введите числа через пробел: 133 4 5 9090 13 12 145

Распределение по количеству цифр:
1-значных: 2
2-значных: 2
3-значных: 2
4-значных: 1
```

Задание 2.

9) Для данного положительного целого числа n напишите функцию, которая возвращает количество установленных битов в его двоичном представлении (также известном как вес Хэмминга).

Input: $n = 11$

Output: 3

Input: $n = 128$

Output: 1

Input: $n = 2147483645$

Output: 30

Выполнение:

Код программы:

```
def hamming_weight(n: int) -> int:
    return bin(n).count("1")

while True:
    user_input = input("Введите положительное целое число: ")
    try:
        n = int(user_input)
        if n < 0:
            print("Ошибка: число должно быть положительным.\n")
            continue
        break
    except ValueError:
        print("Ошибка: нужно ввести целое число.\n")

print("Количество установленных битов:", hamming_weight(n))
```

Спецификация ввода

Введите положительное целое число: <число>

Пример

Введите положительное целое число: 122

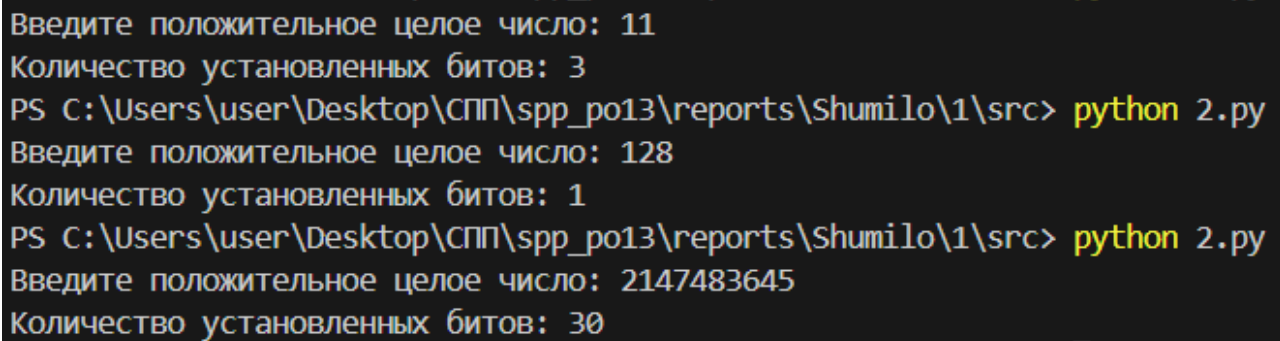
Спецификация вывода

Количество установленных битов: <количество>

Пример

Количество установленных битов: 3

Рисунки с результатами работы программы



```
Введите положительное целое число: 11
Количество установленных битов: 3
PS C:\Users\user\Desktop\СПП\spp_po13\reports\Shumilo\1\src> python 2.py
Введите положительное целое число: 128
Количество установленных битов: 1
PS C:\Users\user\Desktop\СПП\spp_po13\reports\Shumilo\1\src> python 2.py
Введите положительное целое число: 2147483645
Количество установленных битов: 30
```

Вывод: закрепил базовые знания языка программирования Python при решении практических задач.