

10. Функции. Аргументы 1

Ограничение времени	10 секунд
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Допустим, ваша компания каждый месяц выделяет некоторую сумму денег на подарки сотрудникам ко дню рождения. Каждый месяц эта сумма разная. Также у вас есть список дней рождения сотрудников. Вам нужно написать удобный инструмент для того, чтобы сотрудники могли понять, сколько денег им ждать на день рождения, если известно, какой бюджет выделяется на месяц и для какого месяца производится подсчет. Для этого напишите функцию `gift_count`, которая будет принимать:

- бюджет, который выделяется компанией на месяц
- номер месяца, на который нужно произвести расчет
- словарь, где ключи - это имена сотрудников, а значения - `datetime.date` с датой рождения сотрудника.

Функция должна вывести на экран фразу (цифры должны быть подставлены в соответствии со значениями аргументов):

Именинники в месяце 5: Иванов Иван Иванович (01.05.1989), Петров Петр Петрович (06.05.1998). При бюджете 20000 они получают по 10000 рублей.

Если в ответе получается не целое число, округлите его до меньшего целого. Именинники должны быть выведены в порядке дня рождения по возрастанию (не полной даты рождения, а именно дня). Если в выбранном месяце нет именинников, должна быть выведена строка "В этом месяце нет именинников."

Формат ввода

```
birthdays = {"Иванов Иван Иванович": datetime.date(1989, 5, 1), "Петров Петр Петрович": datetime.date(1998, 5, 6)}
gift_count(20000, 5, birthdays)
gift_count(budget=20000, month=5, birthdays=birthdays)
```

Формат вывода

Именинники в месяце 5: Иванов Иван Иванович (01.05.1989), Петров Петр Петрович (06.05.1998). При бюджете 20000 они получают по 10000 рублей.

Примечания

В примере формата ввода показаны несколько вариантов вызова функции `gift_count`: с порядковыми аргументами и с передачей их по имени. Соответственно, аргументы функции должны называться так же, как они используются в примере.

Набрать здесь

Отправить файл

```
1 import datetime
2 from collections import defaultdict
3
4 actual_birthdays = defaultdict()
5
6 def output(month, money, budget, sort_birthdays):
7     txt = "Именинники в месяце " + str(month) + ":"
8
9     for para in sort_birthdays:
10         if para == sort_birthdays[-1:][0]:
11             txt += " " + para[0] + " (" + datetime.datetime.strftime(para[1], "%d.%m.%Y") + ")."
12         else:
13             txt += " " + para[0] + " (" + datetime.datetime.strftime(para[1], "%d.%m.%Y") + "), "
14
15     txt += " При бюджете " + str(budget) + " они получают по " + str(money) + " рублей."
16
17     return txt
18
19
20 def gift_count(budget, month, birthdays):
21     for key, value in birthdays.items():
22         if value.month == month:
23             actual_birthdays[key] = value
24
25     if not actual_birthdays:
26         print("В этом месяце нет именинников.")
27     else:
28         sort_birthdays = sorted(actual_birthdays.items(), key = lambda para: para[1].day)
29         money = int(budget/len(sort_birthdays))
30
31         print(output(month, money, budget, sort_birthdays))
32
33
```

[Отправить](#)[Предыдущая](#)[Следующая](#)