

## **Chapter 8**

# *Date and Time*



# Chapter 8

## Date and Time

In This Chapter:

1. Calendar Module
2. Time Delta

### 8.1 Calendar Module

The calendar module allows you to output calendars and provides additional useful functions for them.

```
class calendar.TextCalendar([firstweekday])
```

This class can be used to generate plain text calendars.

#### Sample Code

```
>>> import calendar
```

```
>>>
```

```
>>> print
```

```
calendar.TextCalendar(firstweekday=6).formatyear(2015)
```

```
                2015
    January          February          March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                1 2 3      1 2 3 4 5 6 7      1 2 3 4 5 6 7
 4 5 6 7 8 9 10      8 9 10 11 12 13 14      8 9 10 11 12 13 14
11 12 13 14 15 16 17 15 16 17 18 19 20 21      15 16 17 18 19 20 21
18 19 20 21 22 23 24 22 23 24 25 26 27 28      22 23 24 25 26 27 28
25 26 27 28 29 30 31      29 30 31
    April          May          June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                1 2 3 4                1 2      1 2 3 4 5 6
 5 6 7 8 9 10 11      3 4 5 6 7 8 9      7 8 9 10 11 12 13
12 13 14 15 16 17 18 10 11 12 13 14 15 16      14 15 16 17 18 19 20
19 20 21 22 23 24 25 17 18 19 20 21 22 23      21 22 23 24 25 26 27
26 27 28 29 30      24 25 26 27 28 29 30      28 29 30
                31
```

July							August							September									
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa			
			1	2	3	4							1			1	2	3	4	5			
5	6	7	8	9	10	11	2	3	4	5	6	7	8	6	7	8	9	10	11	12			
12	13	14	15	16	17	18	9	10	11	12	13	14	15	13	14	15	16	17	18	19			
19	20	21	22	23	24	25	16	17	18	19	20	21	22	20	21	22	23	24	25	26			
26	27	28	29	30	31	23	24	25	26	27	28	29	27	28	29	30							
							30	31															

  

October							November							December						
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3	1	2	3	4	5	6	7			1	2	3	4	5
4	5	6	7	8	9	10	8	9	10	11	12	13	14	6	7	8	9	10	11	12
11	12	13	14	15	16	17	15	16	17	18	19	20	21	13	14	15	16	17	18	19
18	19	20	21	22	23	24	22	23	24	25	26	27	28	20	21	22	23	24	25	26
25	26	27	28	29	30	31	29	30	27	28	29	30	31							

## Task

You are given a date. Your task is to find what the *day* is on that date.

## Input Format

A single line of input containing the space separated month, day and year, respectively

## Output Format

Output the correct day in capital letters.

## Sample Input

```
08 05 2015
```

## Sample Output

```
WEDNESDAY
```

## #Code

```
import calendar
month, day, year = map(int, raw_input().split())
print(calendar.day_name[calendar.weekday(year, month, day)].upper())
```

## 8.2 Time Delta

Timestamps are given in the format:

```
Day dd Mon yyyy hh:mm:ss +xxxx
```

Here +xxxx represents the time zone. See the sample below for details.

### Task

Given 2 timestamps, print the absolute difference (in seconds) between them.

### Input Format

The first line contains T, the number of testcases.

Each testcase contains 2 lines, representing time t1 and time t2.

### Output Format

Print the absolute difference in seconds.

### Constraints

It is guaranteed that the input contains only valid timestamps, and the year can reach up to 3000.

### Sample Input

```
2
Sun 10 May 2015 13:54:36 -0700
Sun 10 May 2015 13:54:36 +0000
Sat 02 May 2015 19:54:36 +0530
Fri 01 May 2015 13:54:36 +0000
```

### Sample Output

```
25200
88200
```

### #Code

```
import datetime
def read_datetime():
    return datetime.datetime.strptime(input(), '%a %d %b %Y %H:%M:%S %z')

T = int(input())
```

```
for _ in range(T):
    t1 = read_datetime()
    t2 = read_datetime()
    delta =int(abs(t1 - t2).total_seconds())
    print (delta)
```