Exceptions

Exceptions

Errors detected during execution are called *exceptions*.

Examples:

ZeroDivisionError

This error is raised when the second argument of a division or modulo operation is zero.

```
>>> a = '1'
>>> b = '0'
>>> print int(a) / int(b)
>>> ZeroDivisionError: integer division or modulo by zero
```

ValueError

This error is raised when a built-in operation or function receives an argument that has the right type but an inappropriate value.

```
>>> a = '1'
>>> b = '#'
>>> print int(a) / int(b)
>>> ValueError: invalid literal for int() with base 10: '#'
```

`To learn more about different built-in exceptions` **click here**`.`

Handling Exceptions

The statements *try* and *except* can be used to handle selected exceptions. A *try* statement may have more than one except clause to specify handlers for different exceptions.

```
#Code
try:
    print 1/0
except ZeroDivisionError as e:
print "Error Code:",e

#Output
Error Code: integer division or modulo by zero
```

Task

You are given two values a and b.
Perform integer division and print a/b.

Input Format

The first line contains T, the number of test cases.
The next T lines each contain the space separated values of a and b.

Constraints

$0 < T < 10$

Output Format

Print the value of a/b.
In the case of *ZeroDivisionError* or *ValueError*, print the error code.

Sample Input

```
3
1 0
2 $
3 1
```

Sample Output

```
Error Code: integer division or modulo by zero
Error Code: invalid literal for int() with base 10: '$'
3
```

Note:

For integer division in **Python 3** use `//`.