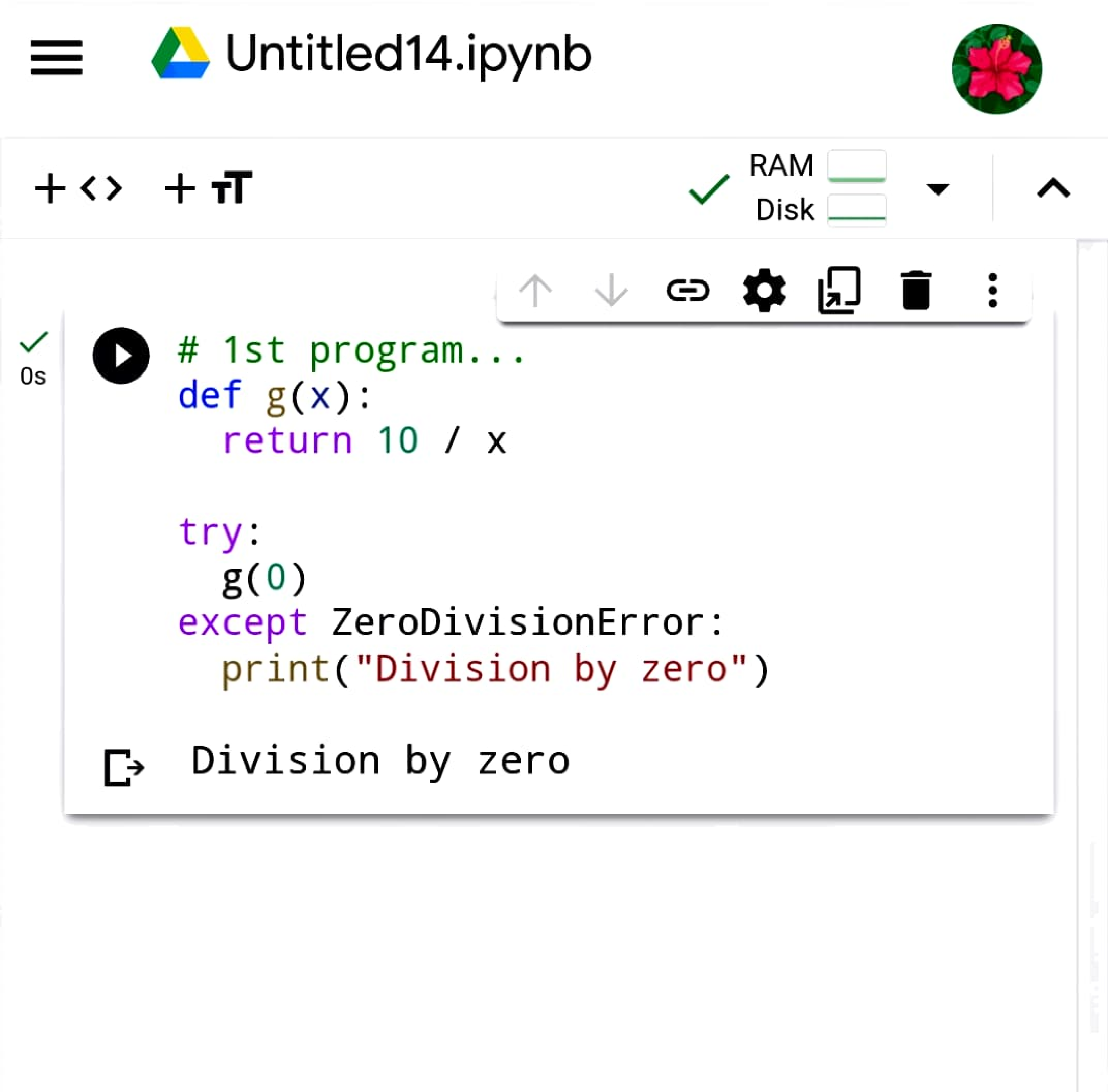
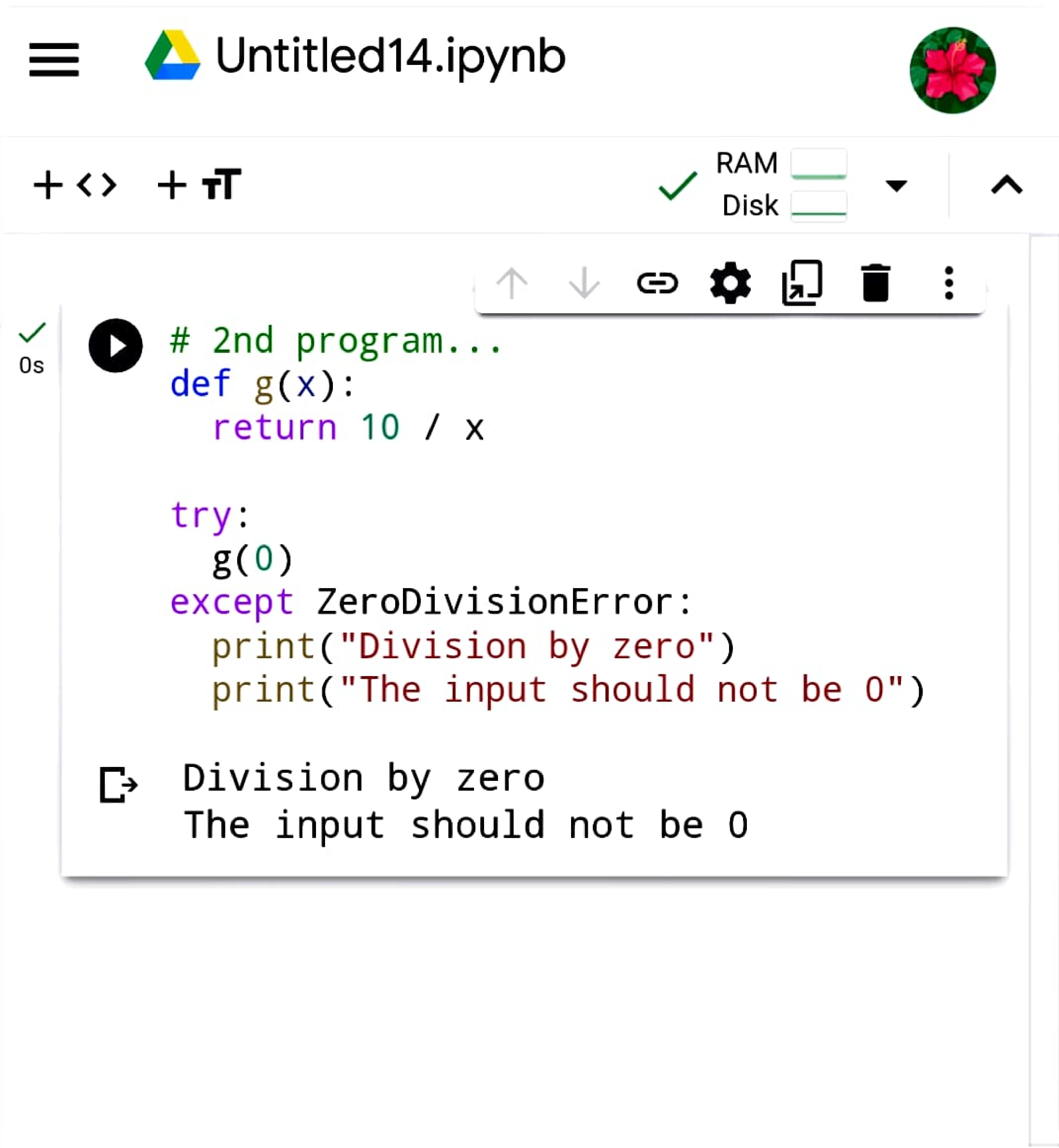
14.Exceptions

1. Write a program to generate Arithmetic Exception without exception handling
2. Handle the Arithmetic exception using try-catch block
3. Write a method which throws exception, Call that method in main class without try block
4. Write a program with multiple catch blocks
5. Write a program to throw exception with your own message
6. Write a program to create your own exception
7. Write a program with finally block
8. Write a program to generate Arithmetic Exception
9. Write a program to generate FileNotFoundException
10. Write a program to generate ClassNotFoundException
11. Write a program to generate IOException
12. Write a program to generate NoSuchFieldException



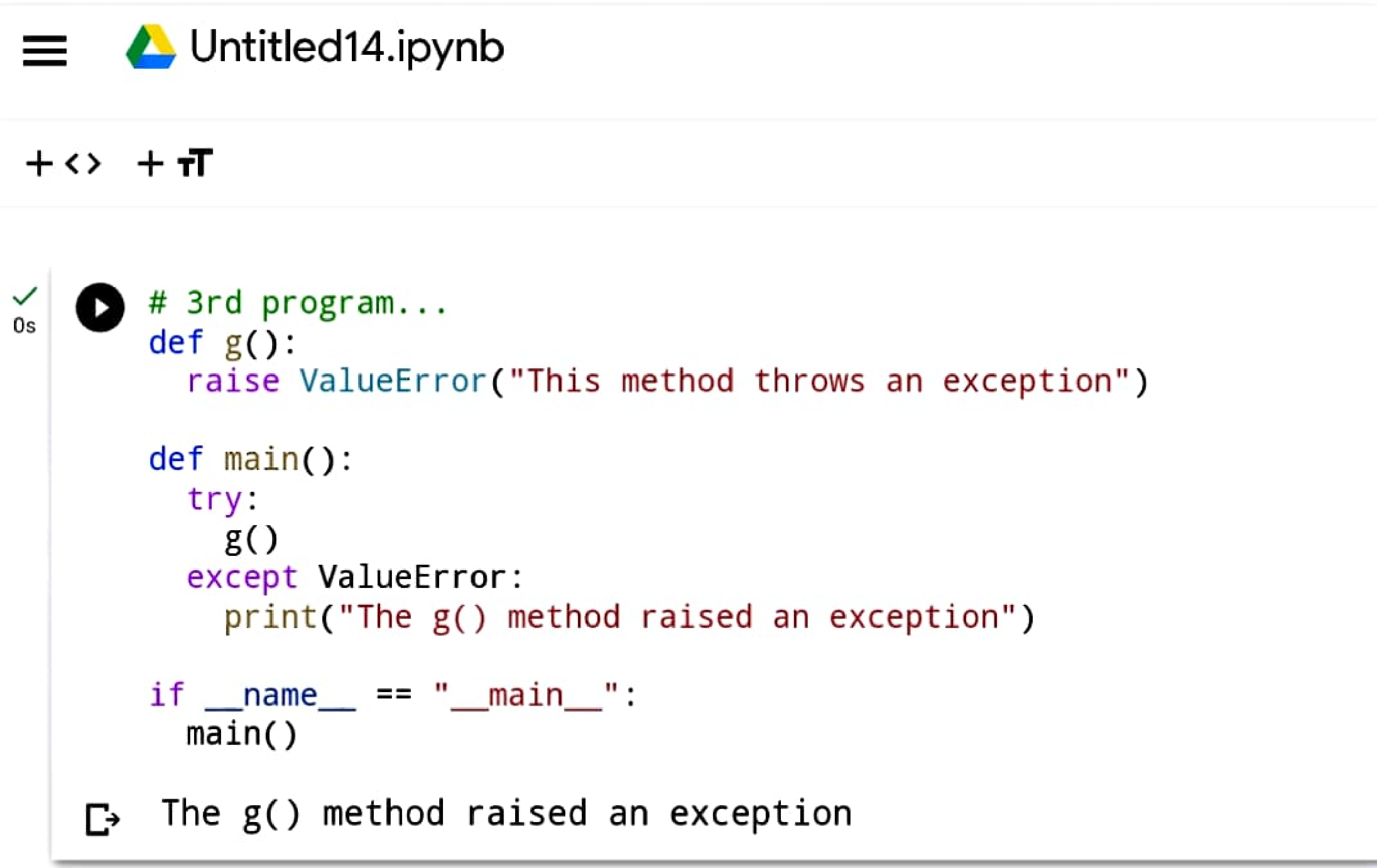
```
# 1st program...
def g(x):
  return 10 / x

try:
  g(0)
except ZeroDivisionError:
  print("Division by zero")
```

```
Division by zero
```



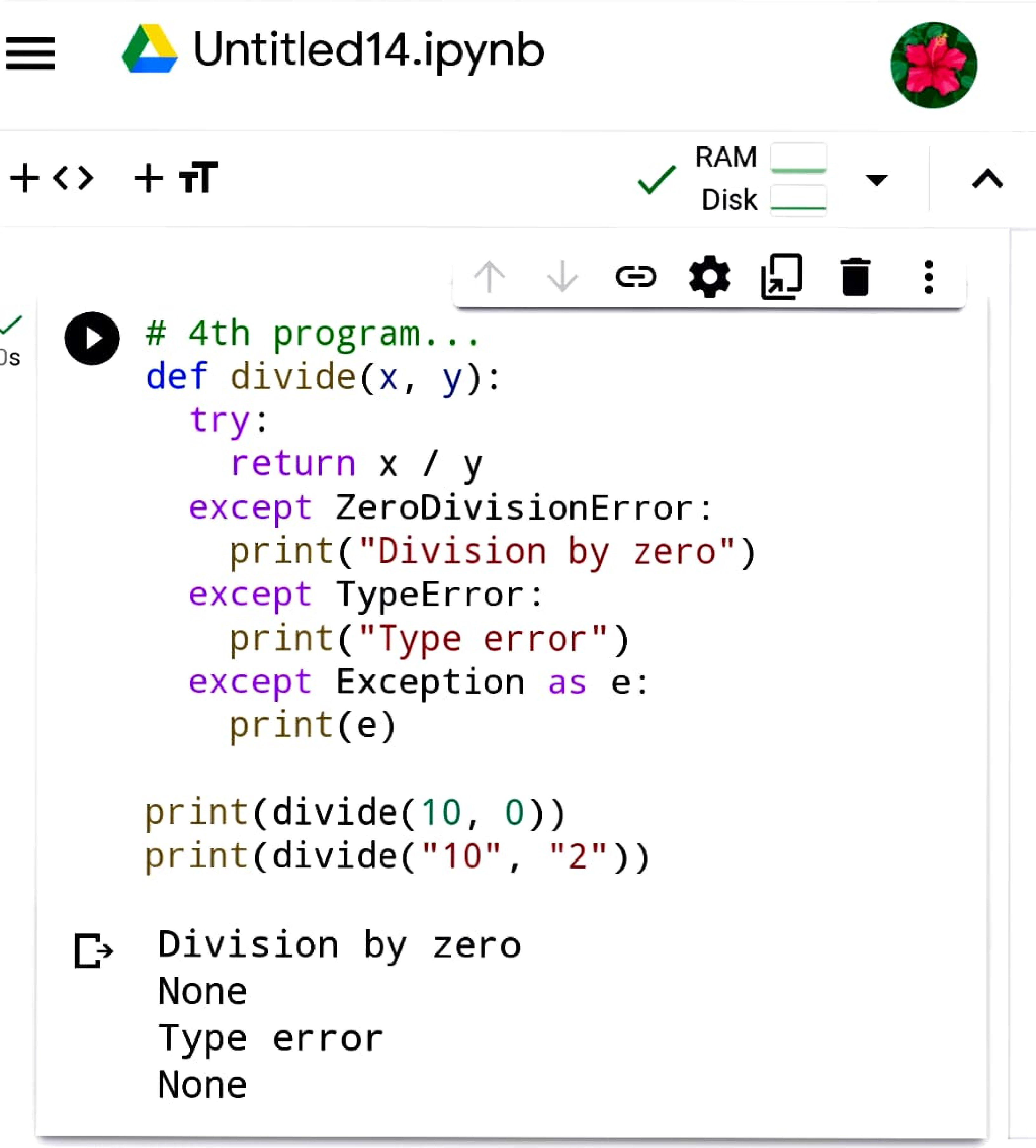
```python
# 2nd program...
def g(x):
  return 10 / x

try:
  g(0)
except ZeroDivisionError:
  print("Division by zero")
  print("The input should not be 0")
```

```
Division by zero
The input should not be 0
```



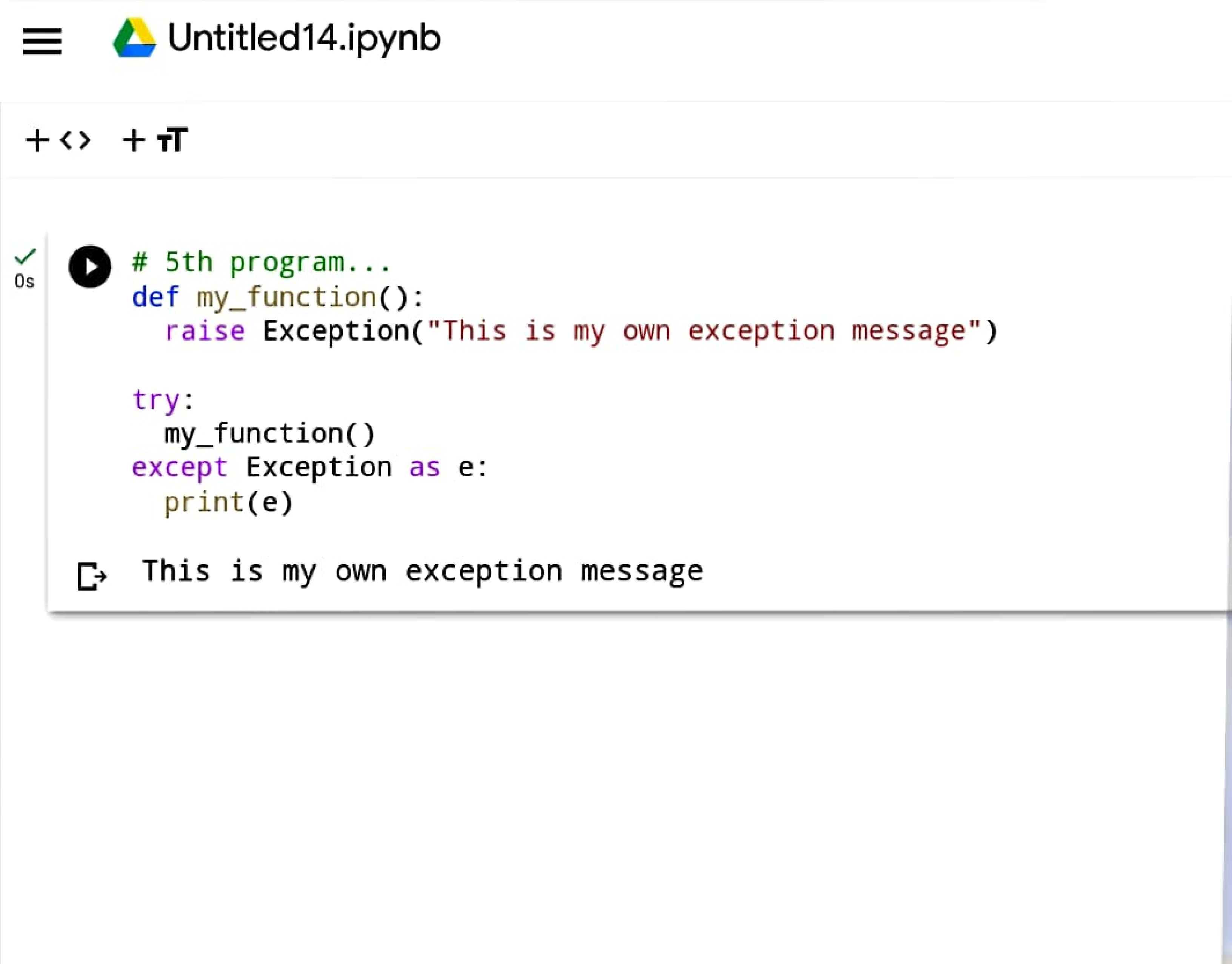
```
# 3rd program...
def g():
  raise ValueError("This method throws an exception")

def main():
  try:
    g()
  except ValueError:
    print("The g() method raised an exception")

if __name__ == "__main__":
  main()
```

```
The g() method raised an exception
```



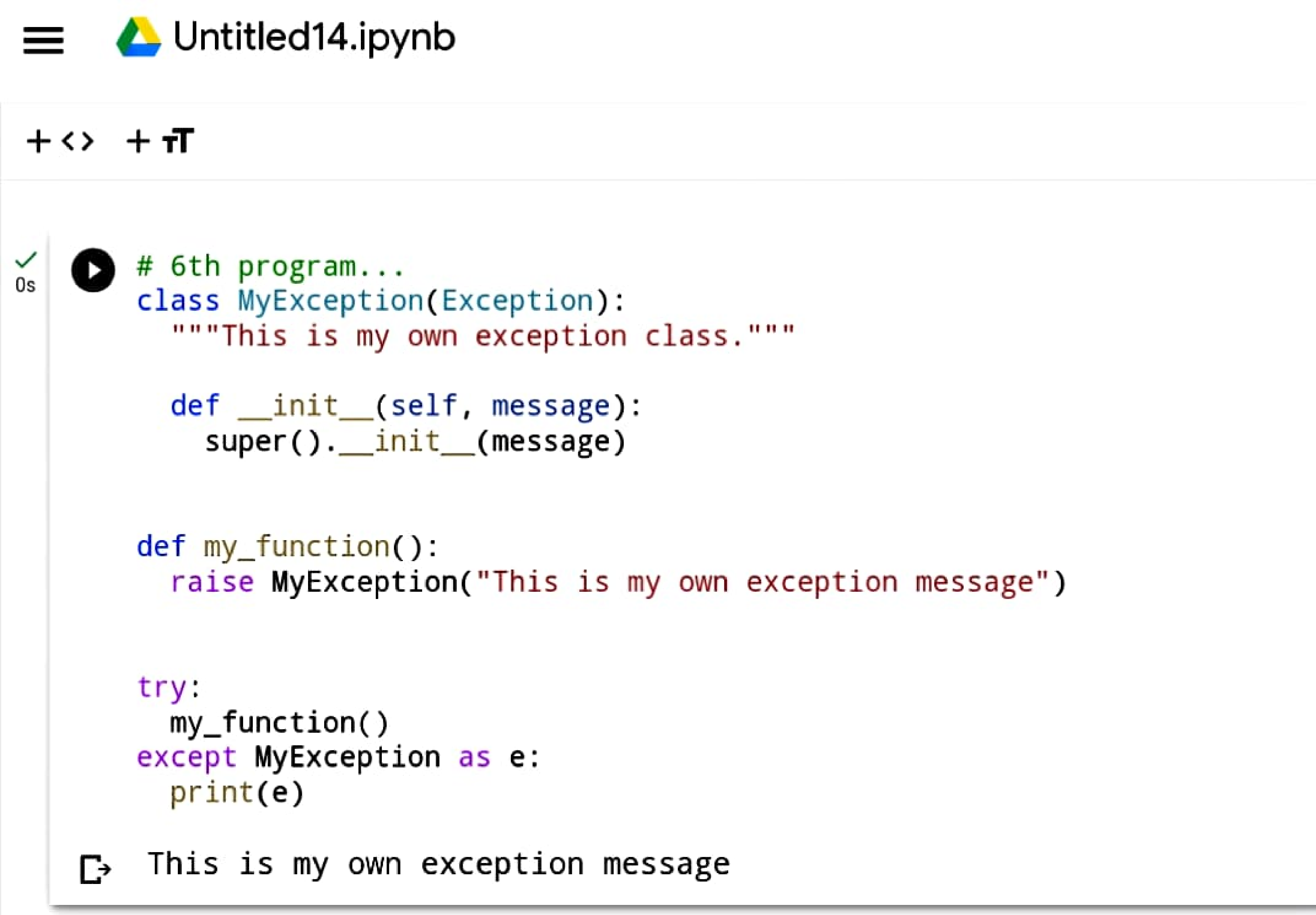
```
# 4th program...
def divide(x, y):
  try:
    return x / y
  except ZeroDivisionError:
    print("Division by zero")
  except TypeError:
    print("Type error")
  except Exception as e:
    print(e)

print(divide(10, 0))
print(divide("10", "2"))
```

```
Division by zero
None
Type error
None
```



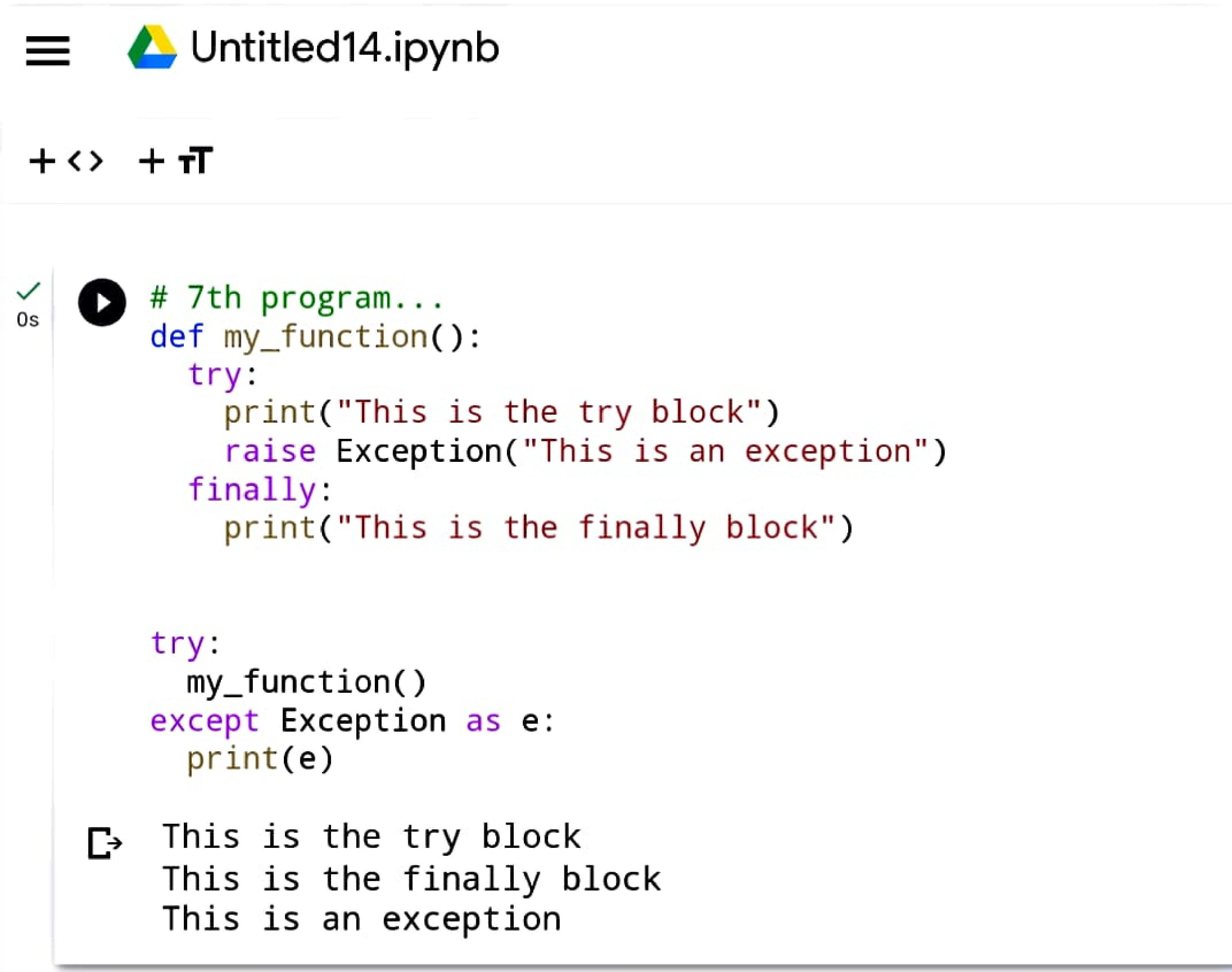
```
# 5th program...
def my_function():
  raise Exception("This is my own exception message")

try:
  my_function()
except Exception as e:
  print(e)
```

```
This is my own exception message
```



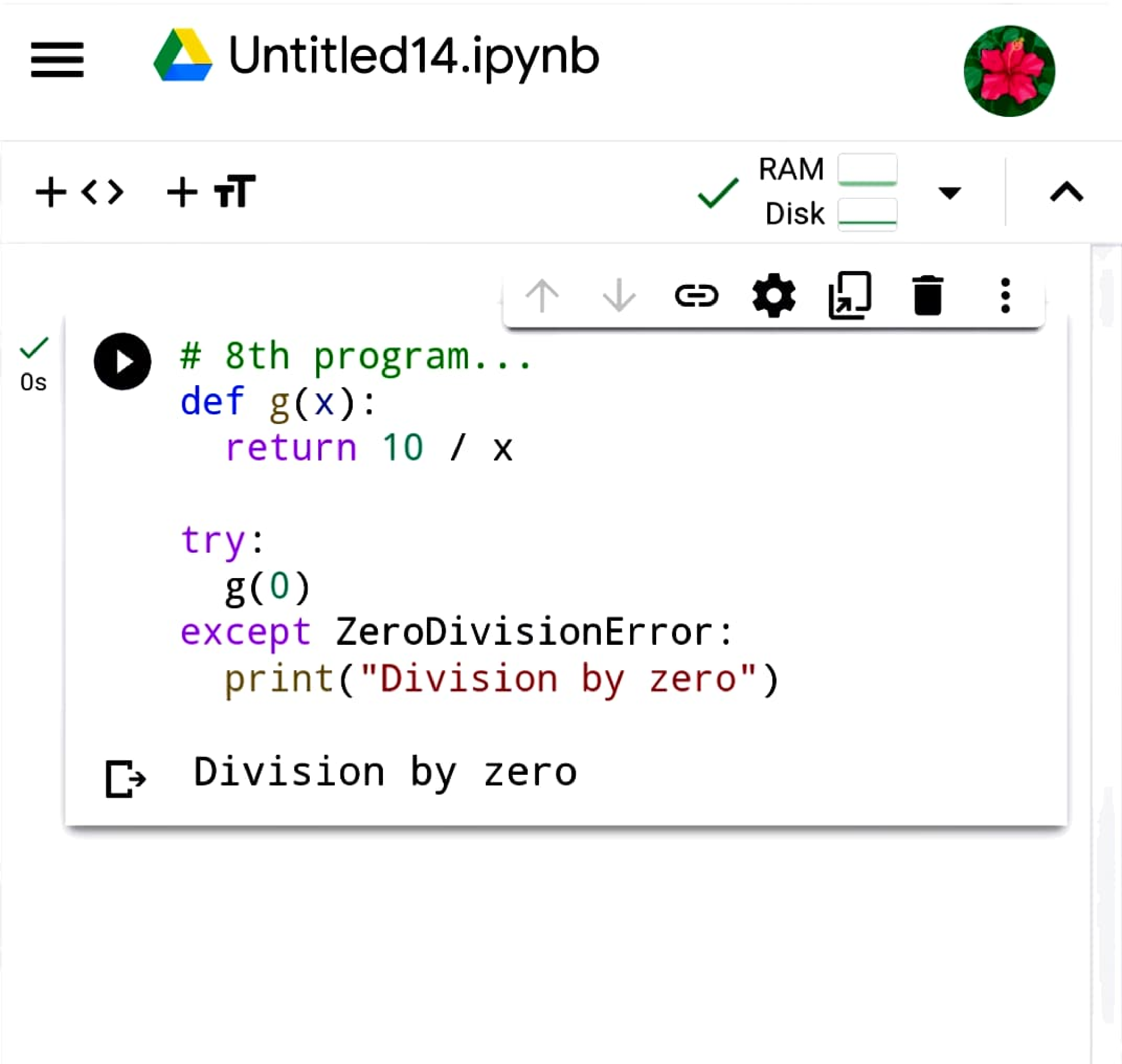
```python
# 6th program...
class MyException(Exception):
  """This is my own exception class."""

  def __init__(self, message):
    super().__init__(message)


def my_function():
  raise MyException("This is my own exception message")


try:
  my_function()
except MyException as e:
  print(e)
```

```
This is my own exception message
```



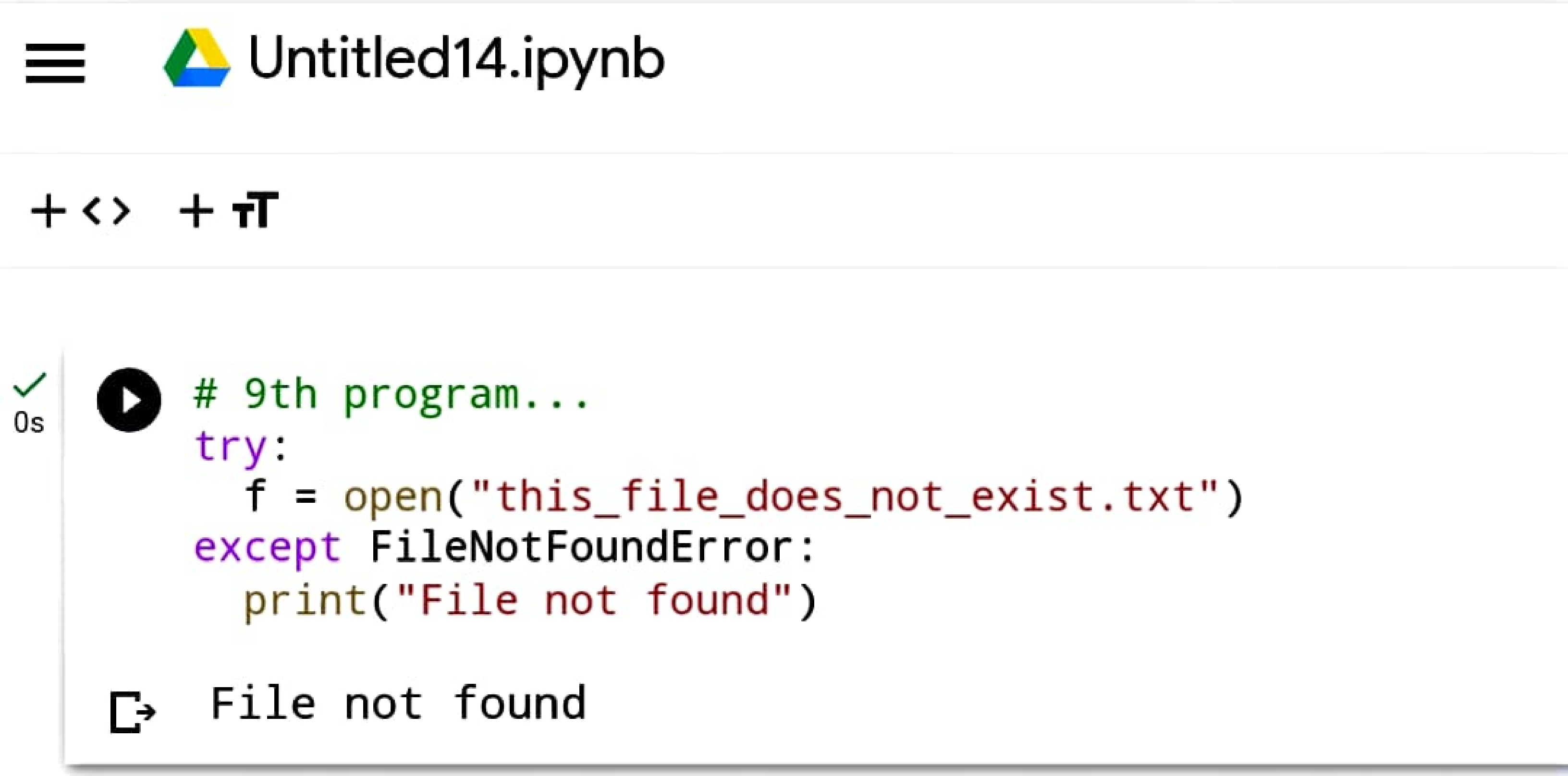
```python
# 7th program...
def my_function():
  try:
    print("This is the try block")
    raise Exception("This is an exception")
  finally:
    print("This is the finally block")


try:
  my_function()
except Exception as e:
  print(e)
```

```
This is the try block
This is the finally block
This is an exception
```



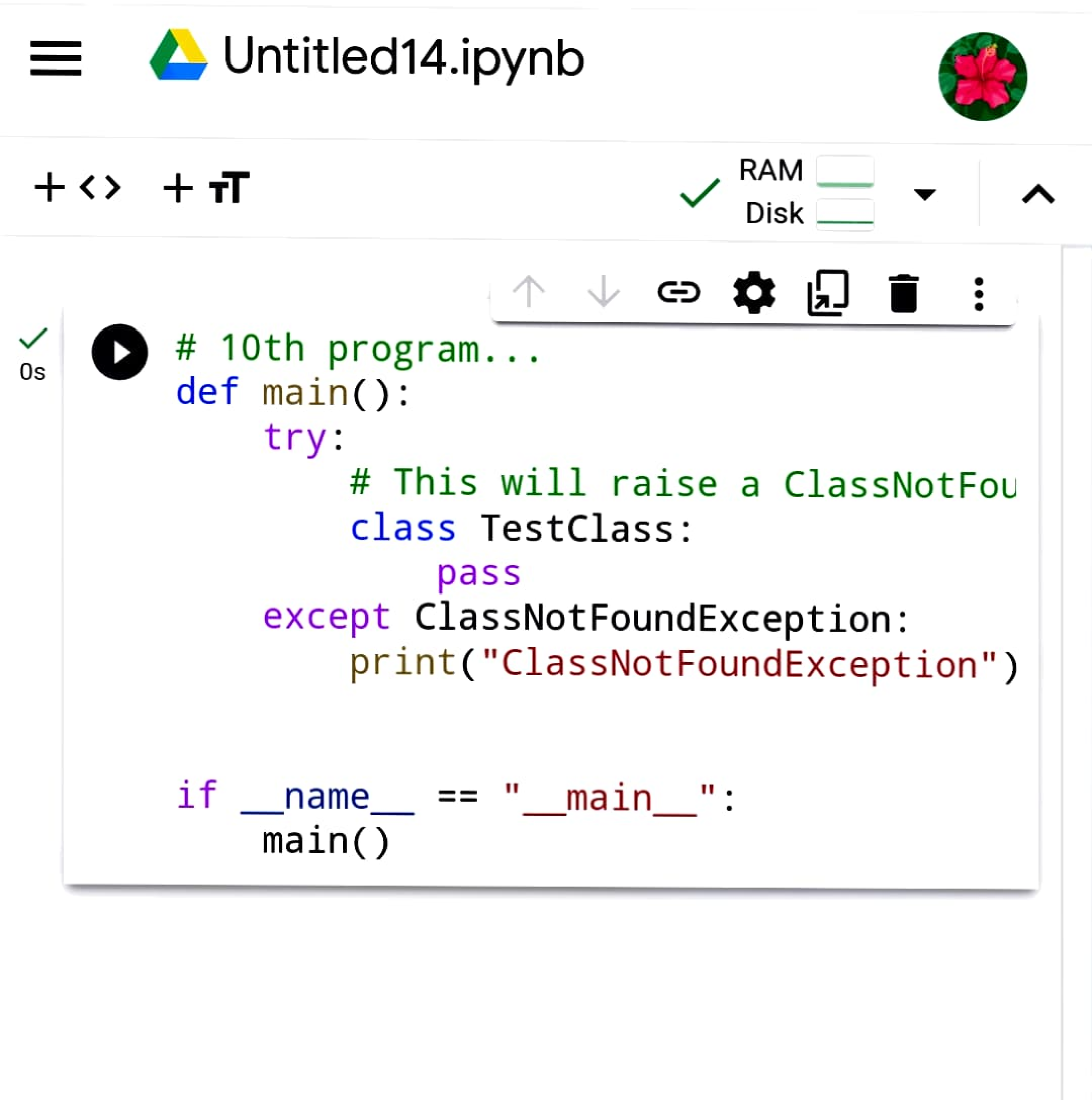
```
# 8th program...
def g(x):
  return 10 / x

try:
  g(0)
except ZeroDivisionError:
  print("Division by zero")
```

```
Division by zero
```

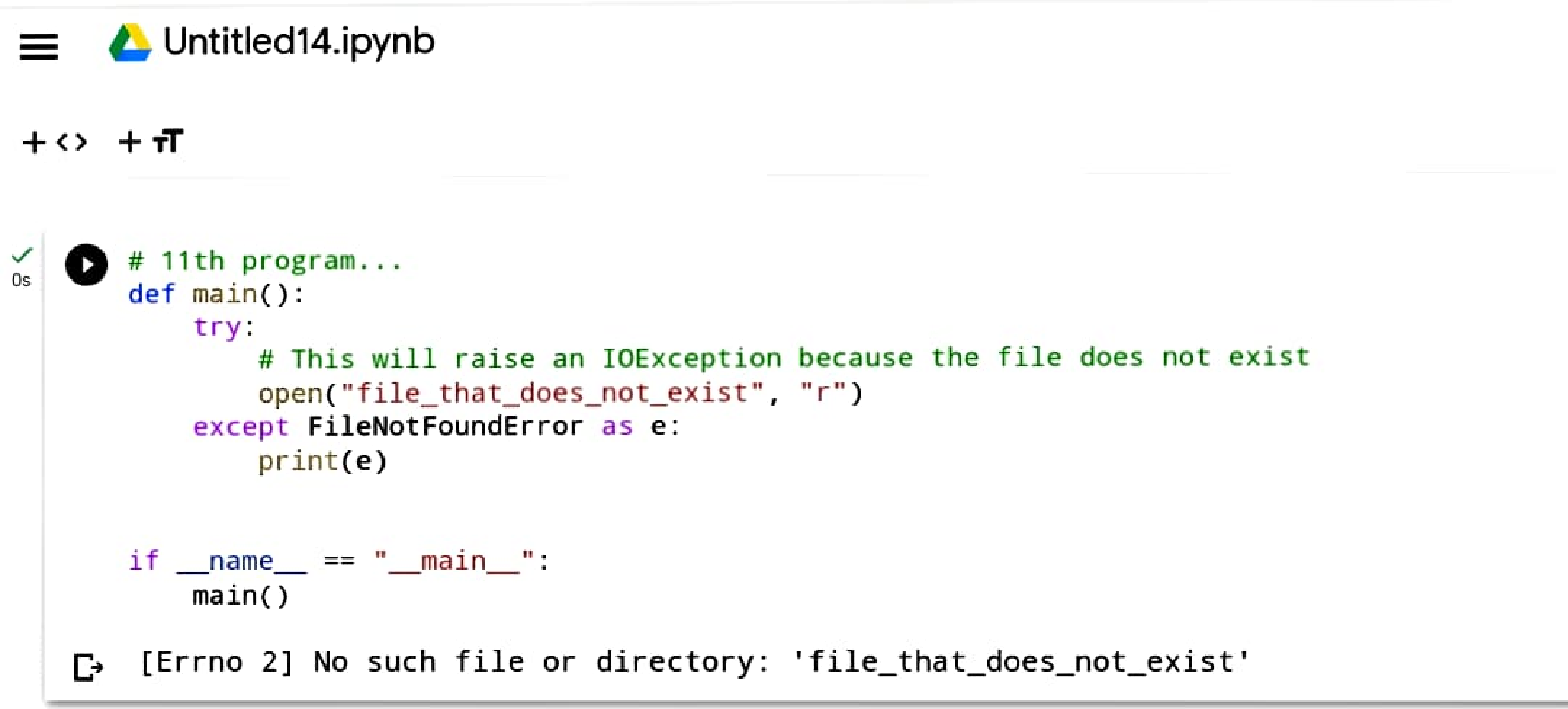


```
# 9th program...
try:
  f = open("this_file_does_not_exist.txt")
except FileNotFoundError:
  print("File not found")
```

```
File not found
```

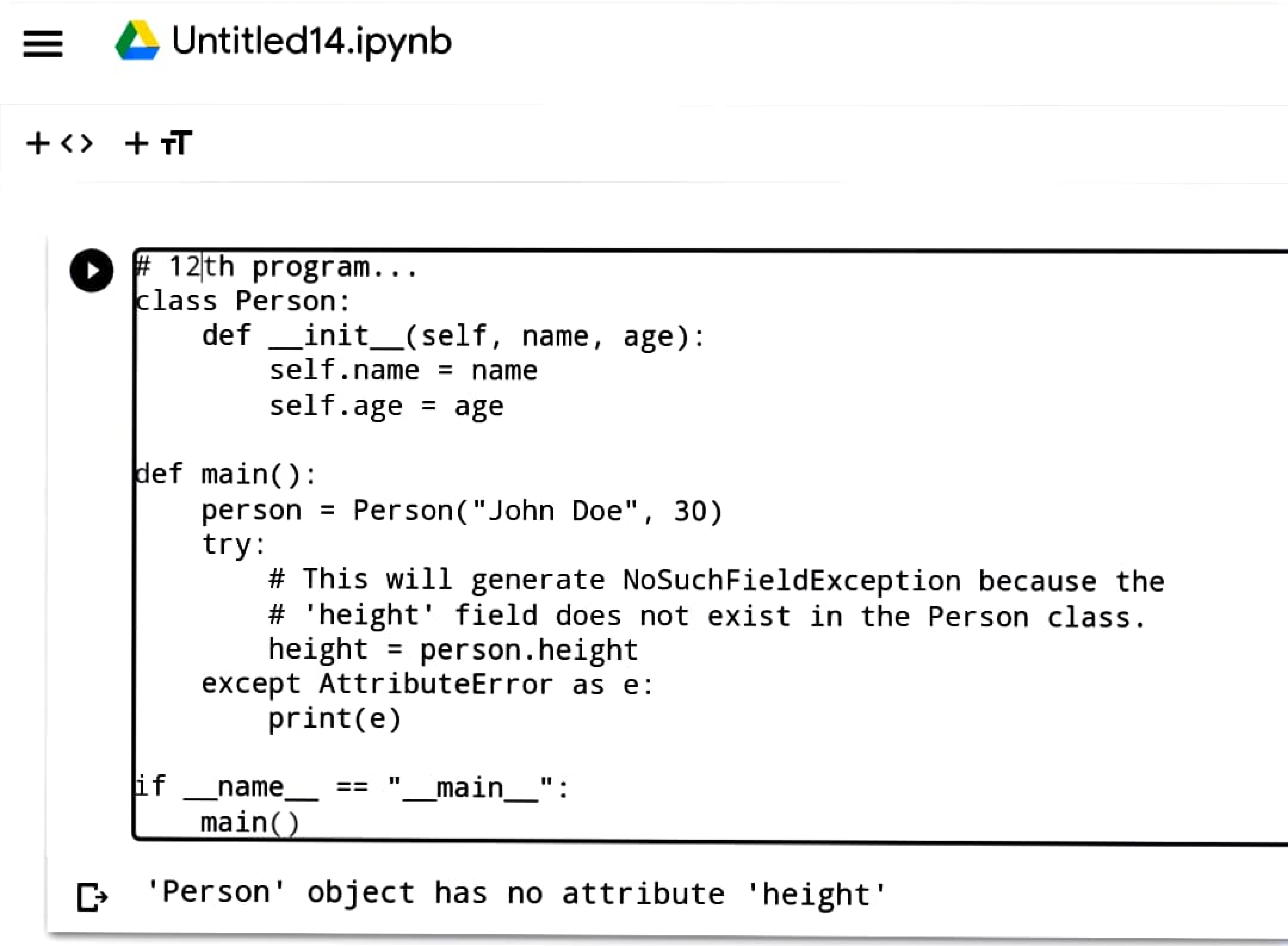


```
# 10th program...
def main():
    try:
        # This will raise a ClassNotFou
        class TestClass:
            pass
    except ClassNotFoundException:
        print("ClassNotFoundException")


if __name__ == "__main__":
    main()
```



```
# 11th program...
def main():
    try:
        # This will raise an IOException because the file does not exist
        open("file_that_does_not_exist", "r")
    except FileNotFoundError as e:
        print(e)


if __name__ == "__main__":
    main()
```

```
[Errno 2] No such file or directory: 'file_that_does_not_exist'
```



```
# 12th program...
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

def main():
    person = Person("John Doe", 30)
    try:
        # This will generate NoSuchFieldException because the
        # 'height' field does not exist in the Person class.
        height = person.height
    except AttributeError as e:
        print(e)

if __name__ == "__main__":
    main()
```

```
'Person' object has no attribute 'height'
```