Command-line usage tutorial for Assignments

In one or more of the questions in your assignments, you are required to input arguments to your program from the command line. Python provides the functionality to parse command-line options and arguments in multiple ways. The most common method is using `sys.argv`.

```
python  programname.py argument1 argument2 argument3
```

```
python3  programname.py argument1 argument2 argument3
```

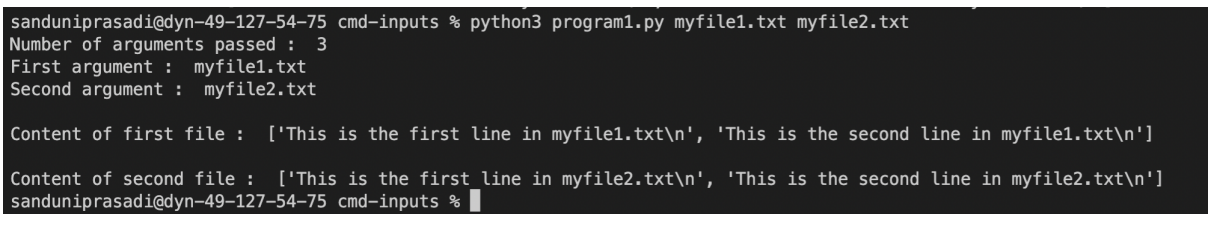
Example 1:

Consider a python program named `program1.py` that takes two files `myfile1.txt` and `myfile2.txt` as inputs to the program:

```python
import sys

# this function reads a file and return its content
def read_file(file_path: str) -> str:
    f = open(file_path, 'r')
    line = f.readlines()
    f.close()

    return line

if __name__ == '__main__':

    #retrieve the file paths from the commandline arguments
    _, filename1, filename2 = sys.argv

    print("Number of arguments passed : ", len(sys.argv))

    # since we know the program takes two arguments
    print("First argument : ", filename1)
    print("Second argument : ", filename2)

    file1content = read_file(filename1)
    print("\nContent of first file : ", file1content)

    file2content = read_file(filename2)
    print("\nContent of second file : ", file2content)
```

Output:

Here we use the terminal to run the python program.



Example 2:

Consider a python program named `program2.py` that takes two files `myfile1.txt` and `myfile2.txt` as inputs to the program. Here we input the absolute paths to the two files in the command line.

```
import sys

# this function reads a file and return its content
def read_file(file_path: str) -> str:
    f = open(file_path, 'r')
    line = f.readlines()
    f.close()

    return line

if __name__ == '__main__':

    print("Number of arguments passed : ", len(sys.argv))

    # this is the program name
    print("0th argument : ", sys.argv[0])
    # first argument/file path
    print("First argument : ", sys.argv[1])
    # second argument/file path
    print("Second argument : ", sys.argv[2])

    print("\nContent of first file : ", read_file(sys.argv[1]))
    print("\nContent of second file : ", read_file(sys.argv[2]))
```

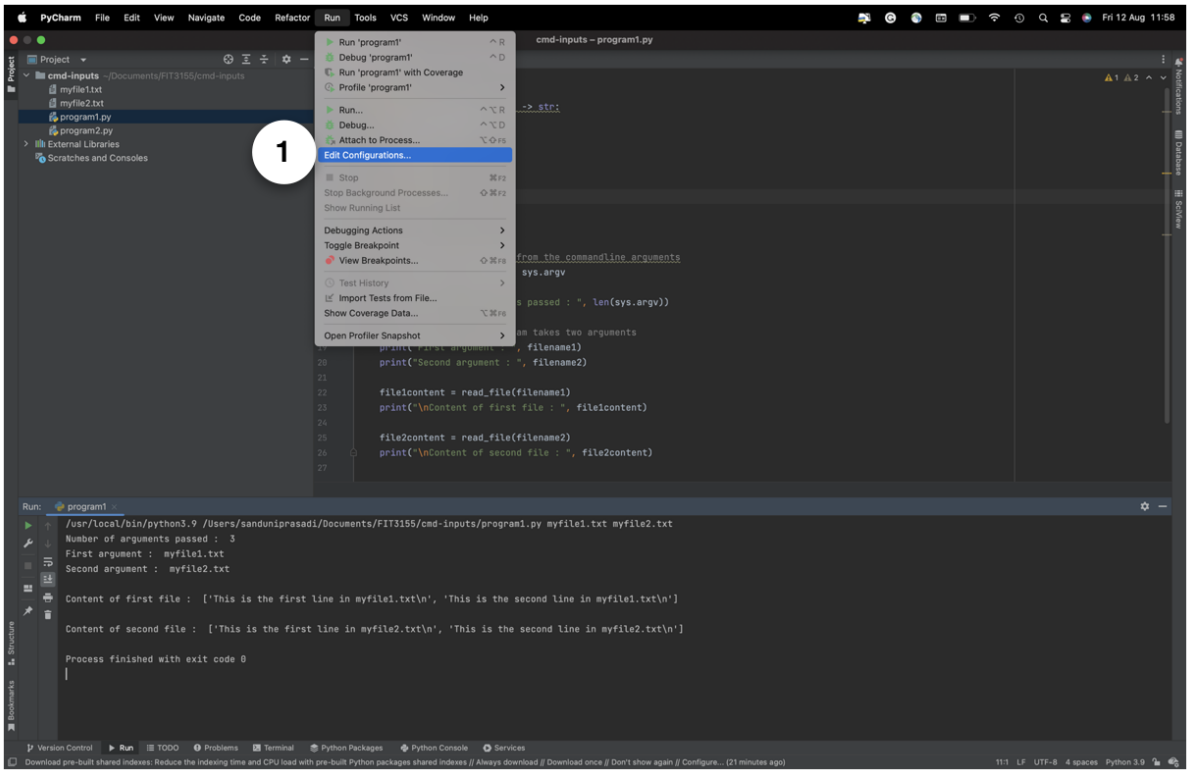
Output:

Here we use the terminal to run the python program.

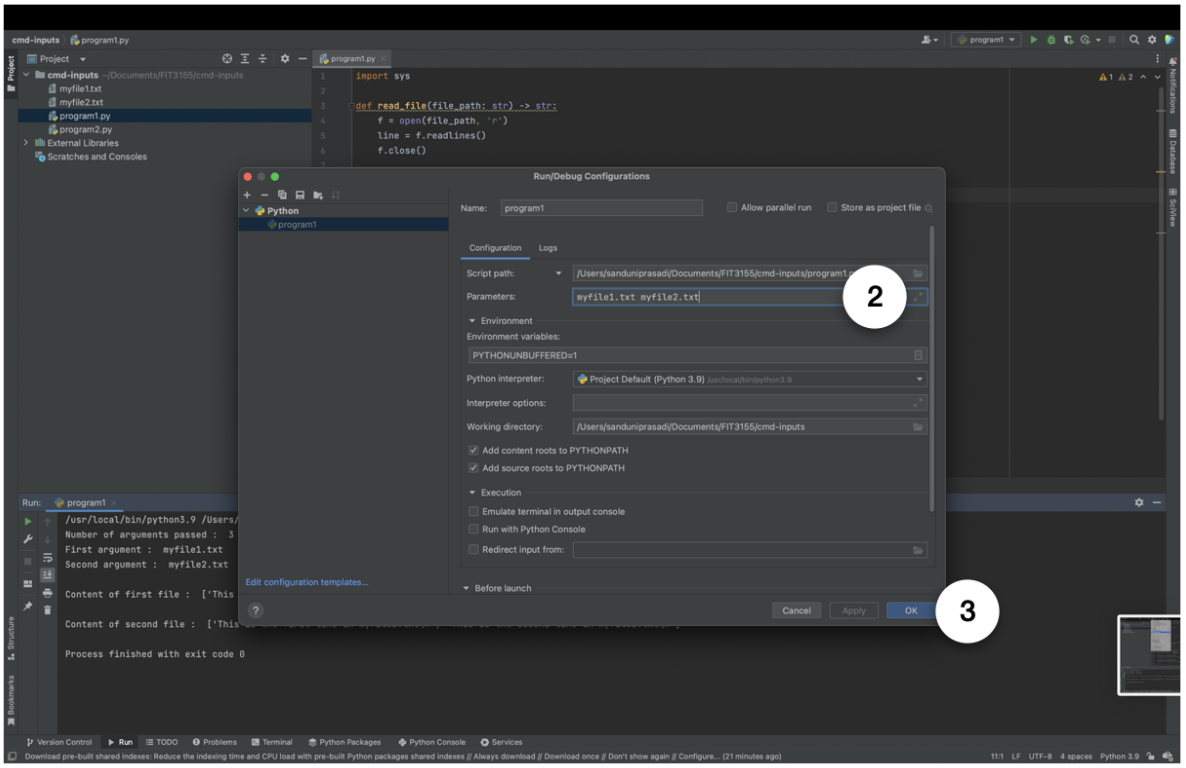
```
sanduniprasadi@dyn-49-127-54-75 cmd-inputs % python3 program2.py ~/Documents/FIT3155/Assignment/myfile1.txt ~/Documents/FIT3155/Assignment/myfile2.txt
Number of arguments passed :  3
0th argument :  program2.py
First argument :  /Users/sanduniprasadi/Documents/FIT3155/Assignment/myfile1.txt
Second argument :  /Users/sanduniprasadi/Documents/FIT3155/Assignment/myfile2.txt

Content of first file :  ['This is the first line in myfile1.txt\n', 'This is the second line in myfile1.txt\n']

Content of second file :  ['This is the first line in myfile2.txt\n', 'This is the second line in myfile2.txt\n']
sanduniprasadi@dyn-49-127-54-75 cmd-inputs %
```

Using PyCharm:

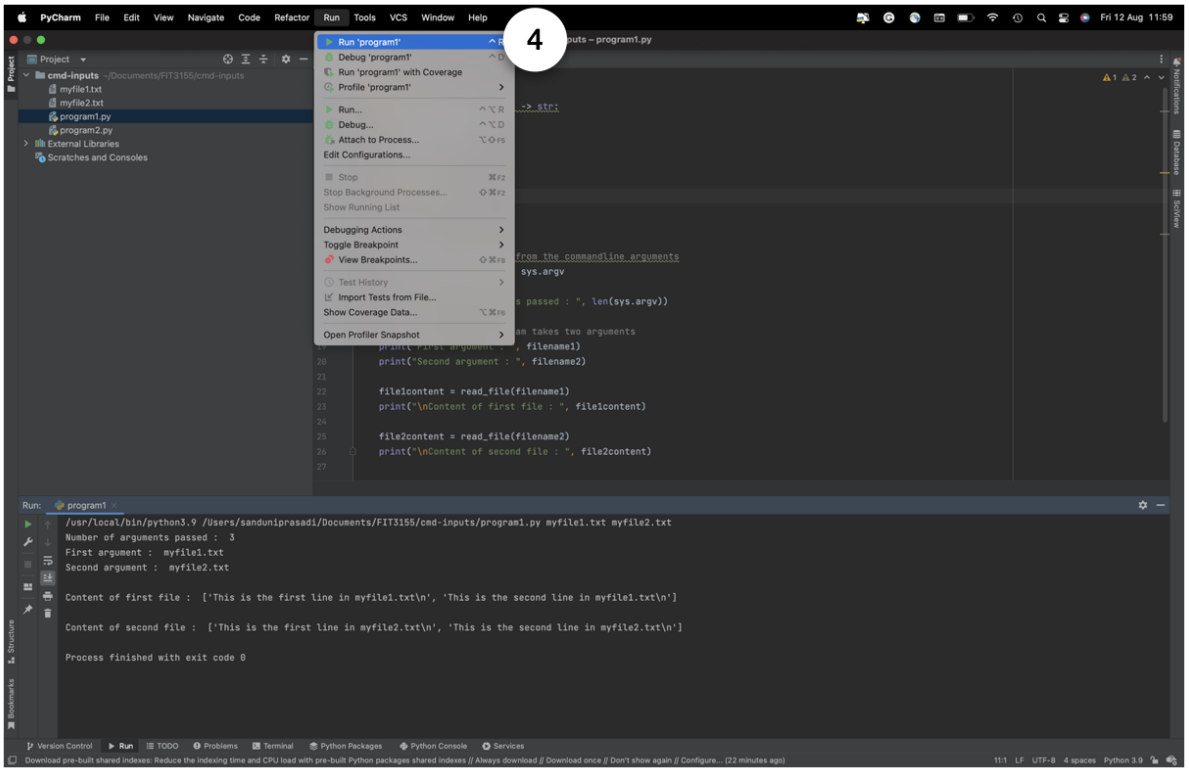
Select Edit configuration from the run tab in the menu list.



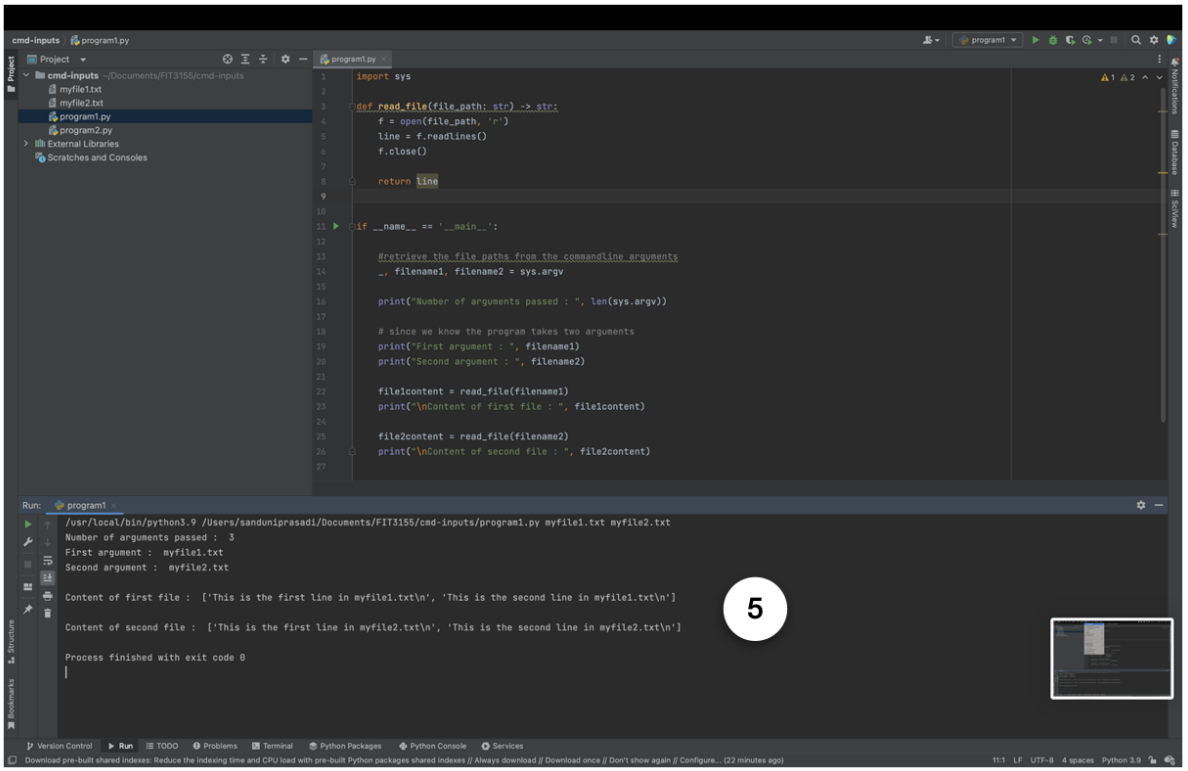
Type the file paths or arguments to your program in the text field next to parameters. Then press OK button.



Run your program.



Output of the `program1.py` is shown below.



The same procedure can be carried out to run the `program2.py`.