

ASSESSMENT–1

Submittedby

NAME :K.SINDHUJA

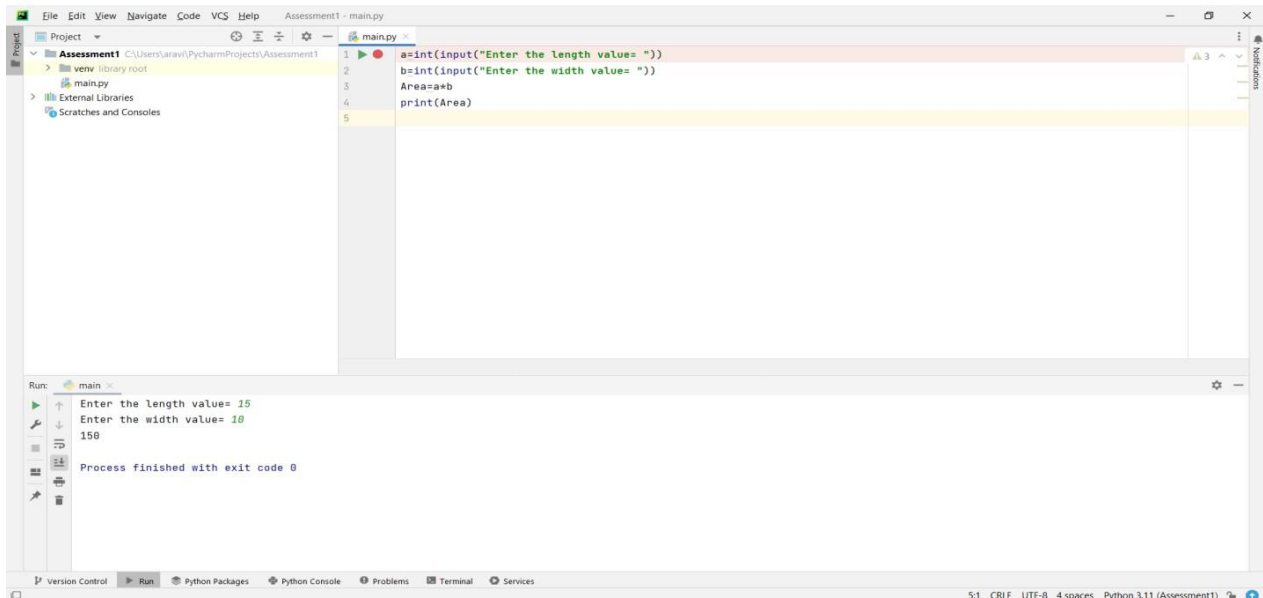
PIN NUMBER :20T91A0533

COLLEGE :GIET ENGINEERING

COLLEGE

E-MAIL. :kadalisindhuja6672@gmail.com

Write a Python program to calculate the area of a rectangle given its length and width. Program :



The screenshot shows a Python IDE with a project named 'Assessment1'. The main.py file contains the following code:

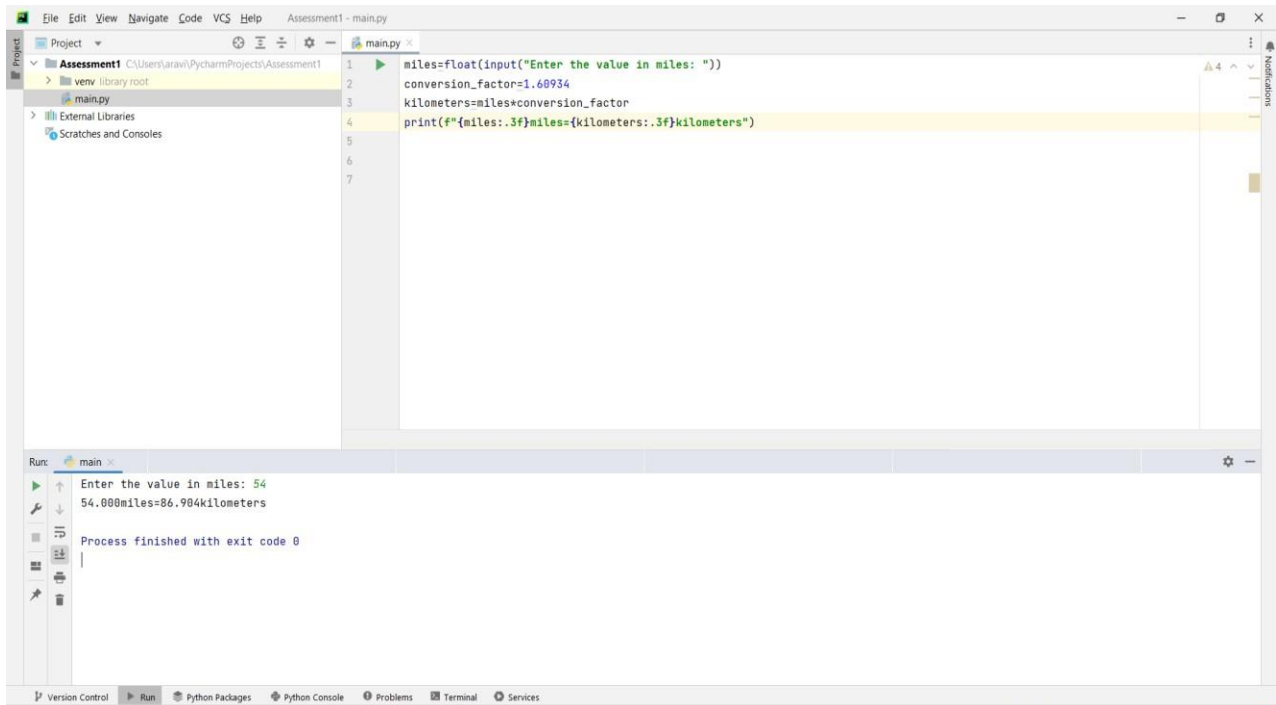
```
1 a=int(input("Enter the length value= "))
2 b=int(input("Enter the width value= "))
3 Area=a*b
4 print(Area)
5
```

The Run console shows the output of the program:

```
Enter the length value= 15
Enter the width value= 10
150
Process finished with exit code 0
```

1. Write a program to convert miles to kilometers.

Program:



The screenshot shows a Python IDE with a project named 'Assessment1'. The main.py file contains the following code:

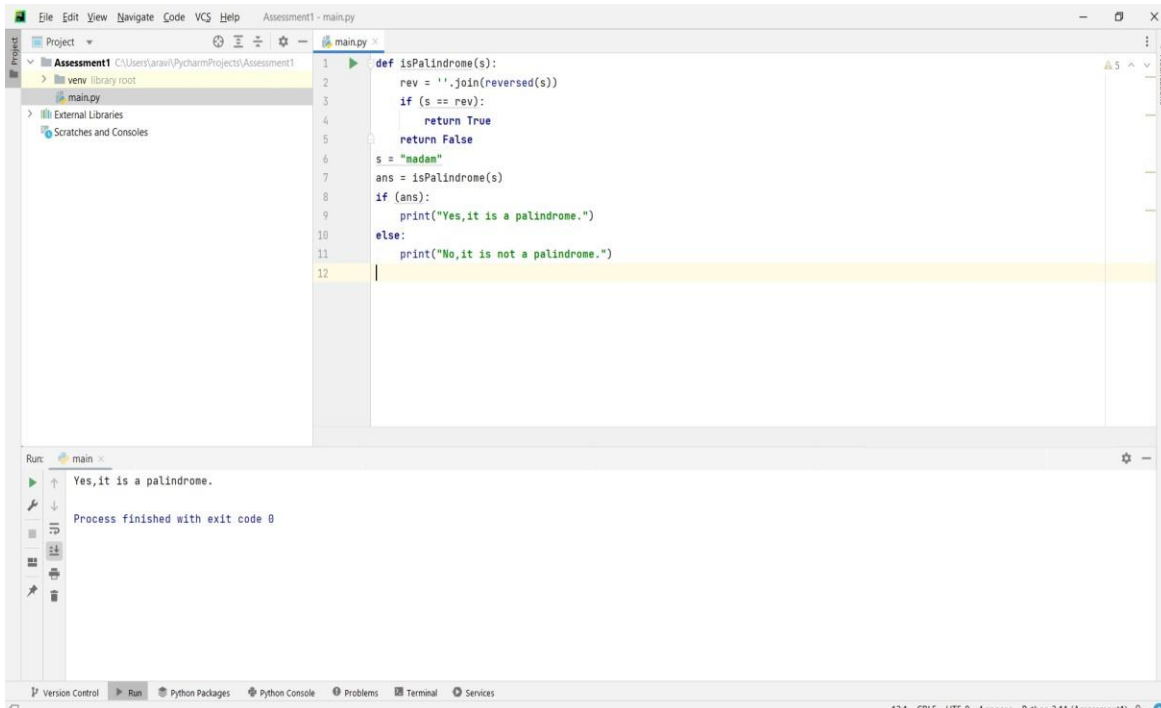
```
1 miles=float(input("Enter the value in miles: "))
2 conversion_factor=1.60934
3 kilometers=miles*conversion_factor
4 print(f"{miles:.3f}miles={kilometers:.3f}kilometers")
5
6
7
```

The Run console shows the output of the program:

```
Enter the value in miles: 54
54.000miles=86.904kilometers
Process finished with exit code 0
```

2. Write a function to check if a given string is a palindrome.

Program :



```
1 def isPalindrome(s):
2     rev = ''.join(reversed(s))
3     if (s == rev):
4         return True
5     return False
6
7 s = "madam"
8 ans = isPalindrome(s)
9 if (ans):
10     print("Yes, it is a palindrome.")
11 else:
12     print("No, it is not a palindrome.")
```

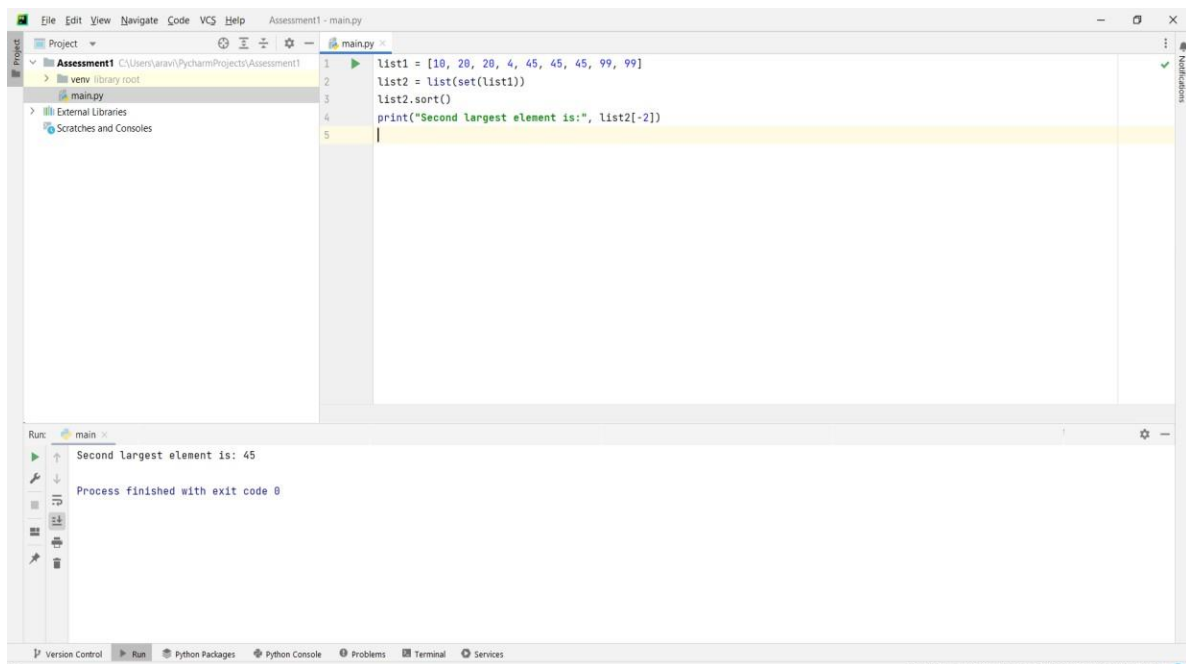
Run: main x

Yes, it is a palindrome.

Process finished with exit code 0

3. Write a Python program to find the second largest element in a list.

Program:



```
1 list1 = [10, 20, 20, 4, 45, 45, 45, 99, 99]
2 list2 = list(set(list1))
3 list2.sort()
4 print("Second largest element is:", list2[-2])
5
```

Run: main x

Second largest element is: 45

Process finished with exit code 0

4. Explain what indentation means in Python.

Ans. In Python, indentation refers to the spaces or tabs at the beginning of a line of code that indicate the block of code it belongs to. It is used to define the structure and hierarchy of code blocks, such as loops, conditional statements, and function definitions. Consistent indentation is crucial in Python because it replaces the need for curly braces or keywords to denote blocks, making the code more readable.

1. Consistency: Python relies heavily on indentation to determine the structure of code. Consistent indentation ensures that the code is organized and easy to understand.

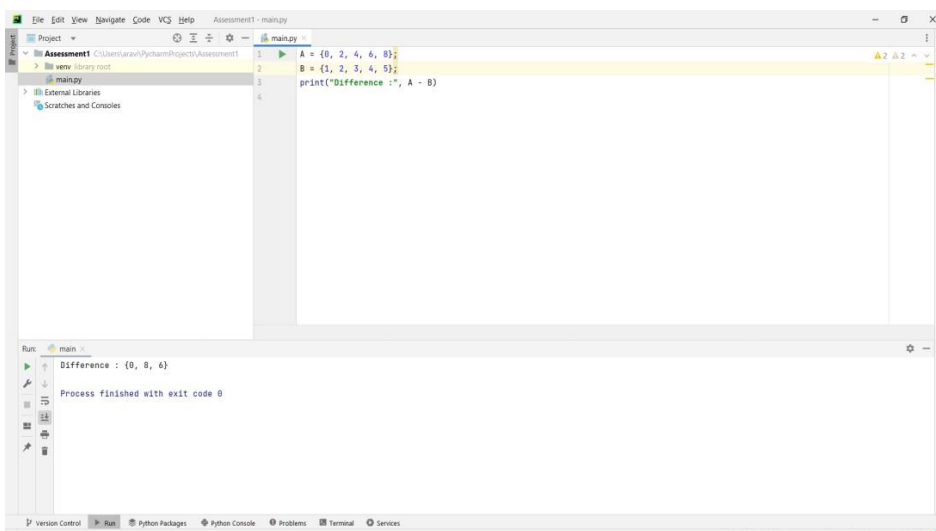
2. Whitespace: Indentation is a form of whitespace in Python. It can be either spaces or tabs, but it's recommended to stick to one style throughout a codebase to avoid confusion.

3. Nested Blocks: Indentation allows for the nesting of code blocks within each other. For example, within a function, you might have nested loops or conditional statements, and their indentation levels indicate their relationship.

4. No Braces: Unlike many other programming languages that use braces { } to denote code blocks, Python uses indentation exclusively. This can sometimes catch new Python developers off guard if they're used to languages with explicit block delimiters.

5. Enforced by Syntax: Incorrect indentation in Python will result in a syntax error. This strict enforcement ensures that code is well-structured and readable.

6. Write a program to perform set difference operation.



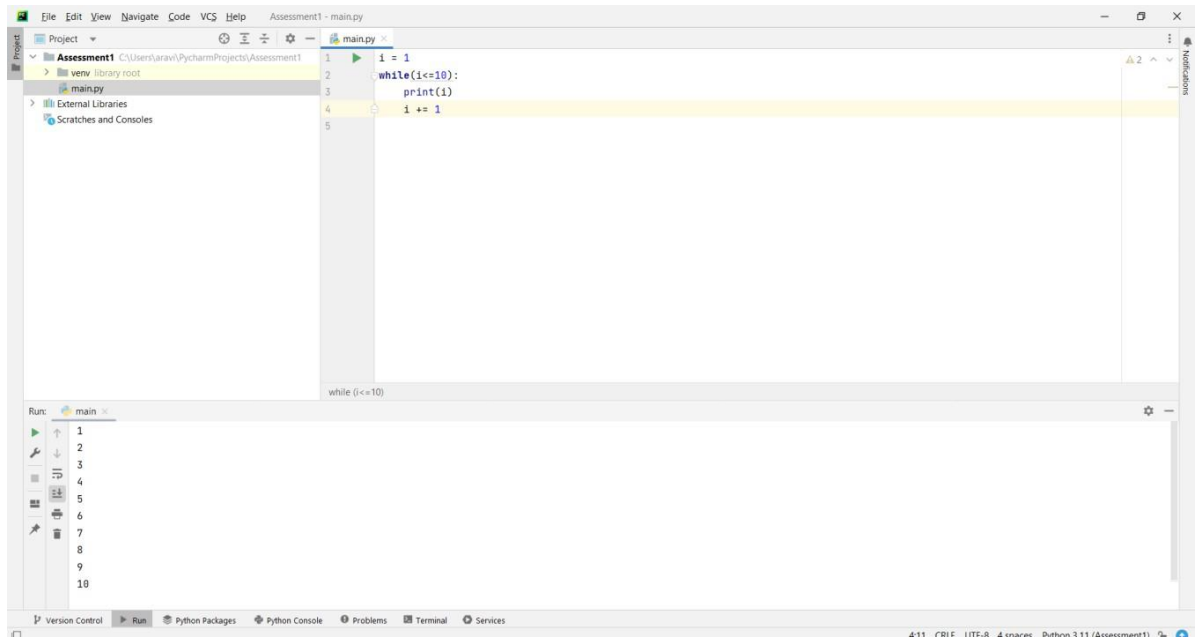
```
1 A = {0, 2, 4, 6, 8}
2 B = {1, 2, 3, 4, 5}
3 print("Difference :", A - B)
4
```

The screenshot shows a Python IDE with a project named 'Assessment1'. The main.py file contains the following code:

```
1 A = {0, 2, 4, 6, 8}
2 B = {1, 2, 3, 4, 5}
3 print("Difference :", A - B)
4
```

The Run console shows the output: 'Difference : {0, 8, 6}'. The process finished with exit code 0.

7. Write a Python program to print numbers from 1 to 10 using a while loop.
Program :

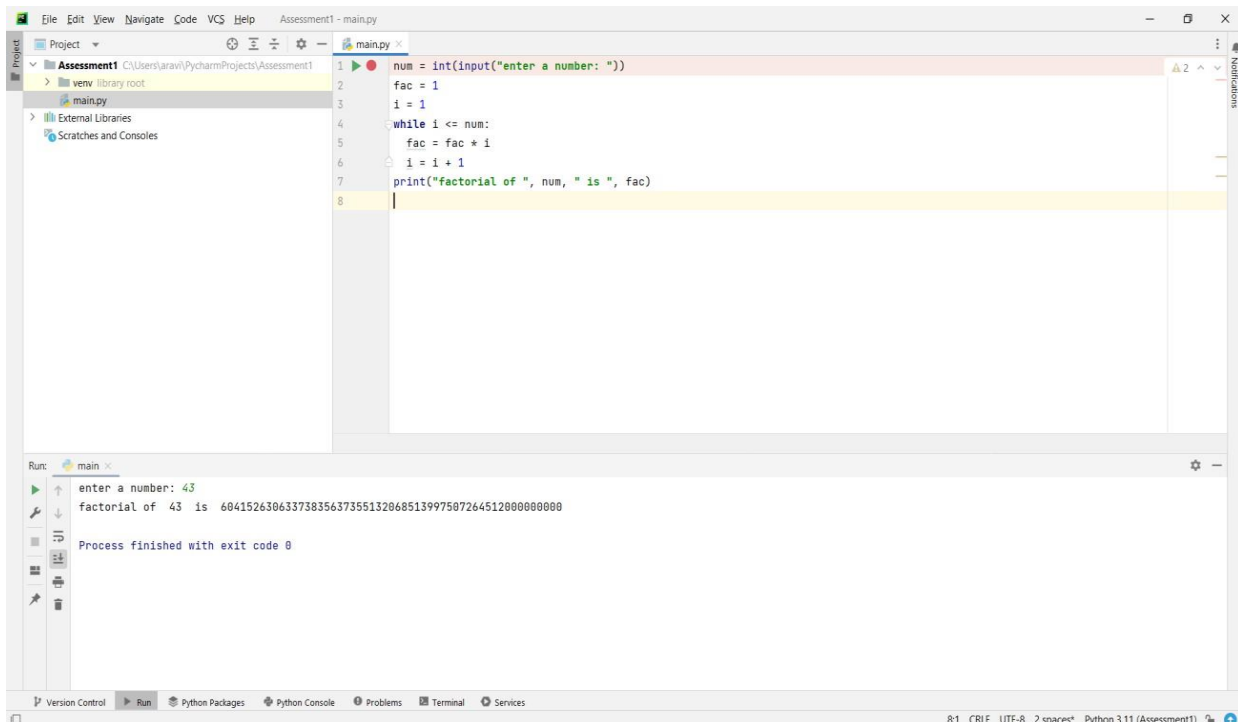


The screenshot shows the PyCharm IDE with a project named 'Assessment1'. The main.py file contains the following code:

```
1 i = 1
2 while(i <= 10):
3     print(i)
4     i += 1
5
```

The Run window at the bottom shows the output of the program, which is the numbers 1 through 10, each on a new line.

8. Write a program to calculate the factorial of a number using a while loop.
Program:



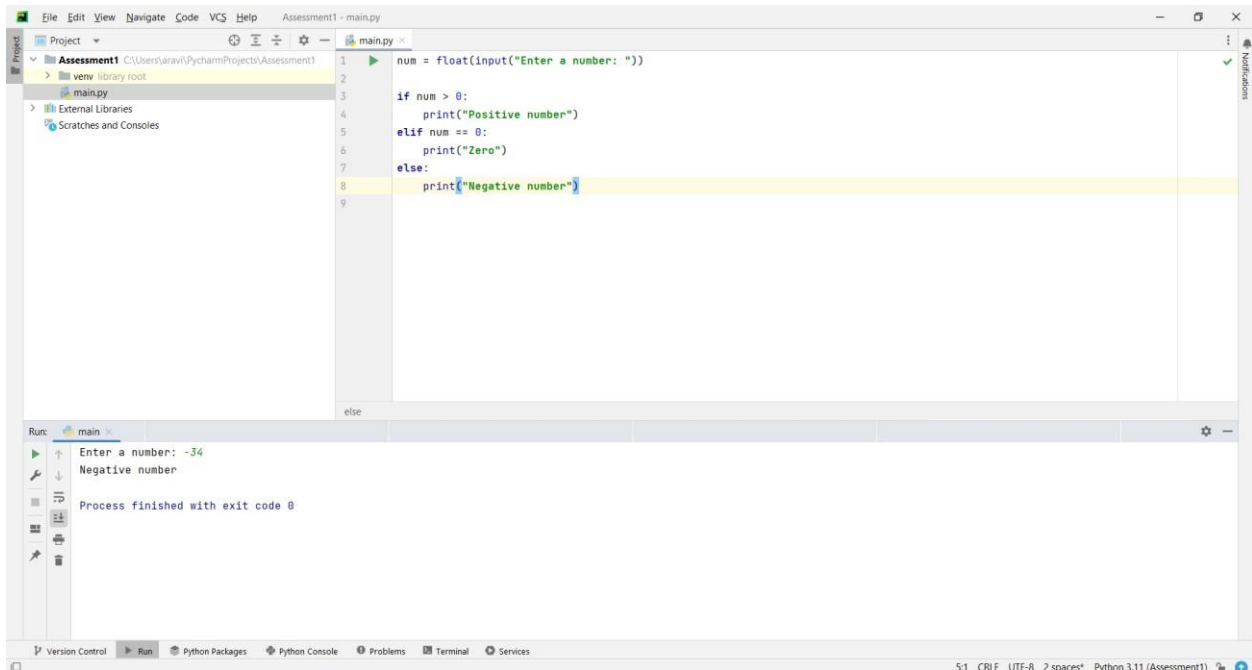
The screenshot shows the PyCharm IDE with a project named 'Assessment1'. The main.py file contains the following code:

```
1 num = int(input("enter a number: "))
2 fac = 1
3 i = 1
4 while i <= num:
5     fac = fac * i
6     i = i + 1
7 print("factorial of ", num, " is ", fac)
8
```

The Run window at the bottom shows the output of the program. The user entered '43', and the program output is 'factorial of 43 is 60415263063373835637355132068513997507264512000000000'. The process finished with exit code 0.

9. Write a Python program to check if a number is positive, negative, or zero using if-elif-else statements.

Program:



The screenshot shows a Python IDE with a project named 'Assessment1'. The main.py file contains the following code:

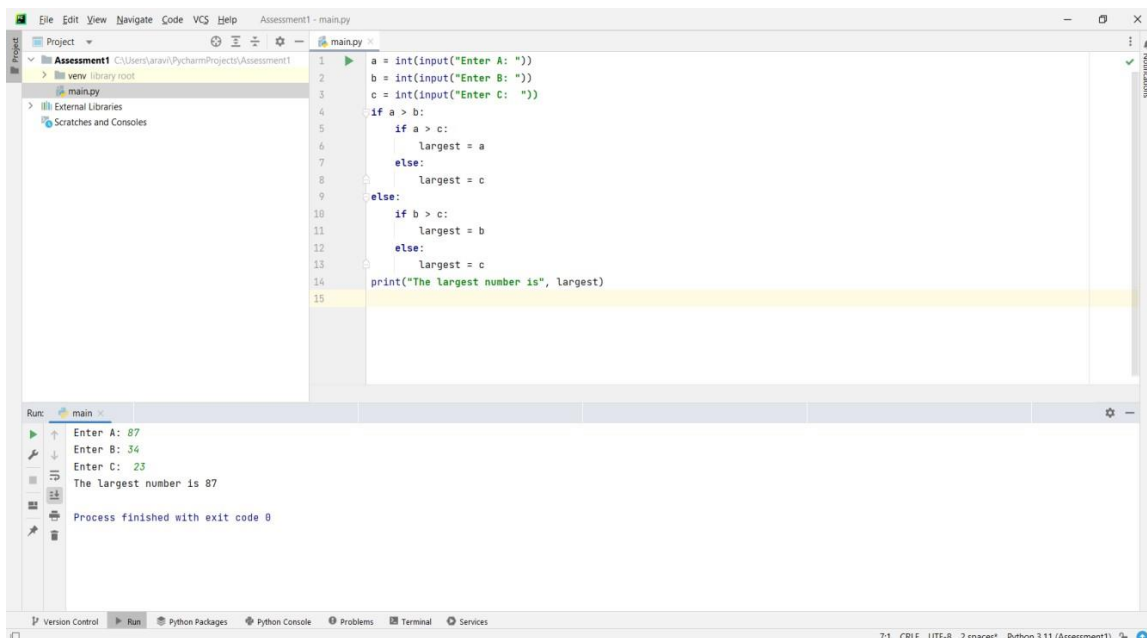
```
1 num = float(input("Enter a number: "))
2
3 if num > 0:
4     print("Positive number")
5 elif num == 0:
6     print("Zero")
7 else:
8     print("Negative number")
9
```

The Run window shows the execution results:

```
Enter a number: -34
Negative number
Process finished with exit code 0
```

10. Write a program to determine the largest among three numbers using conditional statements.

Program :



The screenshot shows a Python IDE with a project named 'Assessment1'. The main.py file contains the following code:

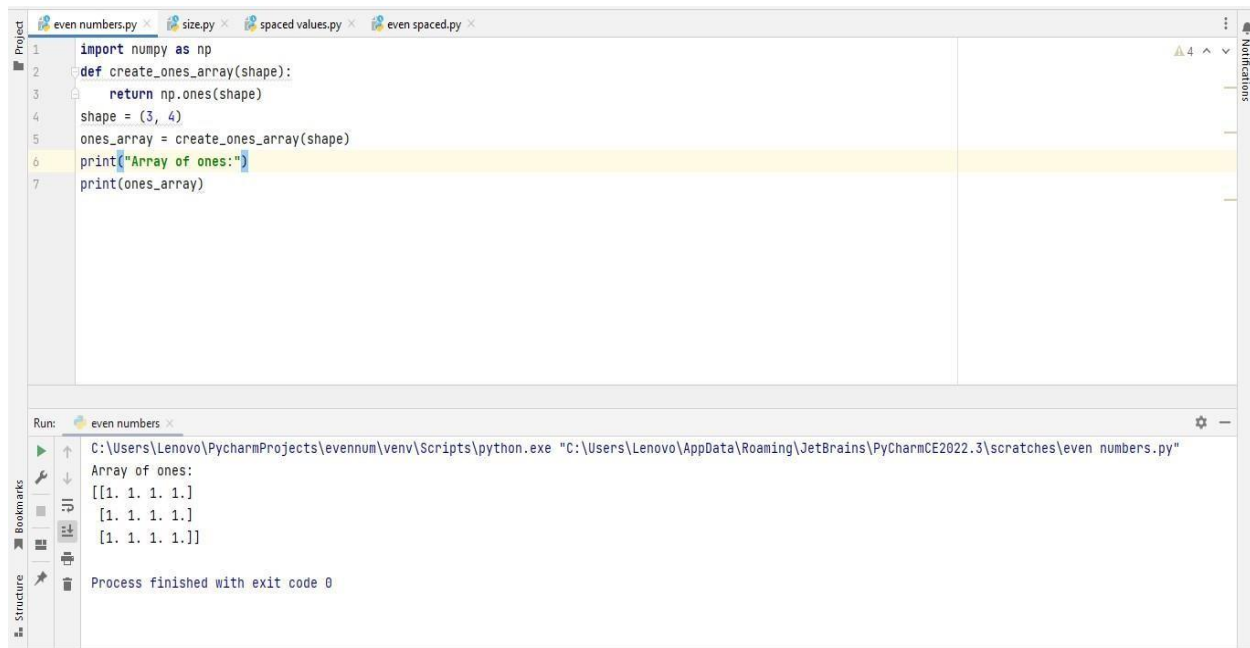
```
1 a = int(input("Enter A: "))
2 b = int(input("Enter B: "))
3 c = int(input("Enter C: "))
4 if a > b:
5     if a > c:
6         largest = a
7     else:
8         largest = c
9 else:
10    if b > c:
11        largest = b
12    else:
13        largest = c
14 print("The largest number is", largest)
15
```

The Run window shows the execution results:

```
Enter A: 87
Enter B: 34
Enter C: 23
The largest number is 87
Process finished with exit code 0
```

11. Write a Python program to create a numpy array filled with ones of given shape.

Program :



The screenshot shows the PyCharm IDE with a project named 'even numbers'. The main editor displays a Python script in 'even numbers.py' with the following code:

```
1 import numpy as np
2 def create_ones_array(shape):
3     return np.ones(shape)
4 shape = (3, 4)
5 ones_array = create_ones_array(shape)
6 print("Array of ones:")
7 print(ones_array)
```

The code is executed, and the Run console shows the output:

```
Run: even numbers
C:\Users\Lenovo\PycharmProjects\evennum\venv\Scripts\python.exe "C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharmCE2022.3\scratches\even numbers.py"
Array of ones:
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
Process finished with exit code 0
```

12. Write a program to create a 2D numpy array initialized with random integers.

Program :



The screenshot shows the PyCharm IDE with a project named 'even numbers'. The main editor displays a Python script in 'even numbers.py' with the following code:

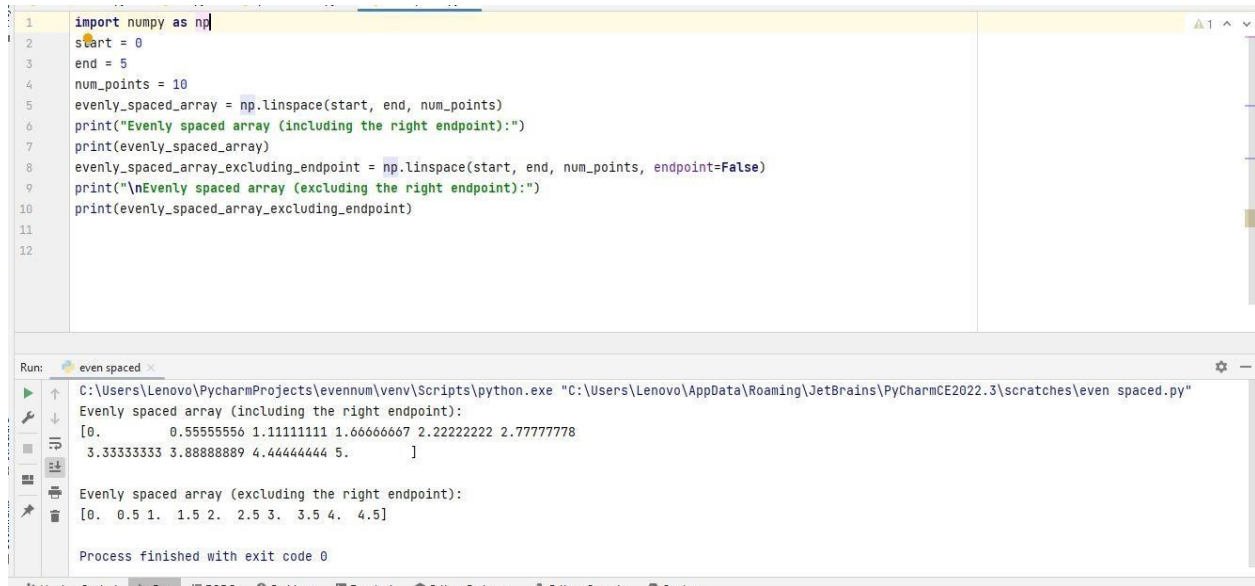
```
1 import numpy as np
2 rows, columns = 3, 4
3 random_array = np.random.randint(10, size=(rows, columns))
4 print("Random 2D array:")
5 print(random_array)
```

The code is executed, and the Run console shows the output:

```
Run: even numbers
C:\Users\Lenovo\PycharmProjects\evennum\venv\Scripts\python.exe "C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharmCE2022.3\scratches\even numbers.py"
Random 2D array:
[[8 3 3 0]
 [8 2 5 1]
 [8 3 5 3]]
Process finished with exit code 0
```

13. Write a Python program to generate an array of evenly spaced numbers over a specified range using `linspace`.

Program:



```
1 import numpy as np
2 start = 0
3 end = 5
4 num_points = 10
5 evenly_spaced_array = np.linspace(start, end, num_points)
6 print("Evenly spaced array (including the right endpoint):")
7 print(evenly_spaced_array)
8 evenly_spaced_array_excluding_endpoint = np.linspace(start, end, num_points, endpoint=False)
9 print("\nEvenly spaced array (excluding the right endpoint):")
10 print(evenly_spaced_array_excluding_endpoint)
11
12
```

Run: even spaced

C:\Users\Lenovo\PycharmProjects\evennum\venv\Scripts\python.exe "C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharmCE2022.3\scratches\even spaced.py"

Evenly spaced array (including the right endpoint):

```
[0. 0.55555556 1.11111111 1.66666667 2.22222222 2.77777778
 3.33333333 3.88888889 4.44444444 5.]
```

Evenly spaced array (excluding the right endpoint):

```
[0. 0.5 1. 1.5 2. 2.5 3. 3.5 4. 4.5]
```

Process finished with exit code 0

14. Write a program to generate an array of 10 equally spaced values between 1 and 100 using `linspace`.

Program:



```
1 import numpy as np
2 equally_spaced_values = np.linspace(1, 100, num=10)
3 print(equally_spaced_values)
4
```

Run: spaced values

C:\Users\Lenovo\PycharmProjects\evennum\venv\Scripts\python.exe "C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharmCE2022.3\scratches\spaced values.py"

```
[ 1. 12. 23. 34. 45. 56. 67. 78. 89. 100.]
```

Process finished with exit code 0

Package numpy installed (49 minutes ago)

15. Write a Python program to create an array containing even numbers from 2 to 20 using `arange`.

Program:



The screenshot shows the PyCharm IDE with a file named `even numbers.py`. The code in the editor is as follows:

```
1 import numpy as np
2 even_numbers = np.arange(2, 21, 2)
3 print("Array of even numbers from 2 to 20:", even_numbers)
4
```

The Run window at the bottom shows the execution output:

```
Run: even numbers
C:\Users\Lenovo\PycharmProjects\evennum\venv\Scripts\python.exe "C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharmCE2022.3\scratches\even numbers.py"
Array of even numbers from 2 to 20: [ 2  4  6  8 10 12 14 16 18 20]
Process finished with exit code 0
```

16. Write a program to create an array containing numbers from 1 to 10 with a step size of 0.5 using `arange`.

Program:



The screenshot shows the PyCharm IDE with two files: `even numbers.py` and `size.py`. The code in `size.py` is as follows:

```
1 import numpy as np
2 numbers = np.arange(1, 10.5, 0.5)
3 print(numbers)
4
```

The Run window at the bottom shows the execution output:

```
Run: size
C:\Users\Lenovo\PycharmProjects\evennum\venv\Scripts\python.exe C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharmCE2022.3\scratches\size.py
[ 1.  1.5  2.  2.5  3.  3.5  4.  4.5  5.  5.5  6.  6.5  7.  7.5
 8.  8.5  9.  9.5 10.]
Process finished with exit code 0
```

