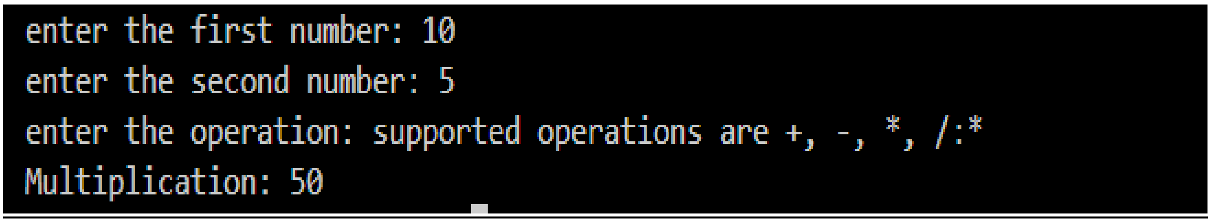
1. Python Programming

1.1. Create a simple calculator in Python.

```
first_no=int(input('enter the first number: '))


second_no=int(input('enter the second number: '))


operation=input('enter the operation: supported operations are +, -, *, /:')


if operation=='+':
   print("Addition:",first_no + second_no)
elif operation=='-':
   print('Substraction:',first_no - second_no)
elif operation=='*':
   print('Multiplication:',first_no * second_no)
elif operation=='/':
   if second_no != 0:
      print('Division:',first_no / second_no)
   else:
      print("Error: Division by zero is not allowed")
else:
   print("Invalid operation")
```

output



```
enter the first number: 10
enter the second number: 5
enter the operation: supported operations are +, -, *, /:*
Multiplication: 50
```

1.2. An electric power distribution company charges domestic customers as follows: Consumption unit Rate of charge:

1.2.1. 0-200 Rs. 0.50 per unit

1.2.2. 201-400 Rs. 0.65 per unit in excess of 200

1.2.3. 401-600 Rs 0.80 per unit excess of 400

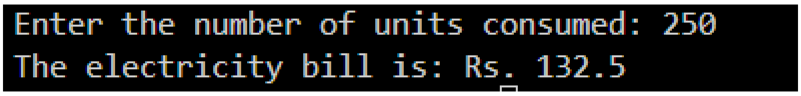
1.2.4. 601 and above Rs 1.00per unit excess of 600

1.2.5. If the bill exceeds Rs. 400, then a surcharge of 15% will be charged, and the minimum bill should be Rs. 100/-

Create a Python program based on the scenario mentioned above.

```
def calculate_electricity_bill(units):
    total_bill = 0

    if units <= 200:
        total_bill = units * 0.50
    elif units <= 400:
        total_bill = 200 * 0.50 + (units - 200) * 0.65
    elif units <= 600:
        total_bill = 200 * 0.50 + 200 * 0.65 + (units - 400) * 0.80
    else:
        total_bill = 200 * 0.50 + 200 * 0.65 + 200 * 0.80 + (units - 600) * 1.00

    if total_bill > 400:
        total_bill *= 1.15

    if total_bill < 100:
        total_bill = 100

    return total_bill

units = int(input("Enter the number of units consumed: "))
```

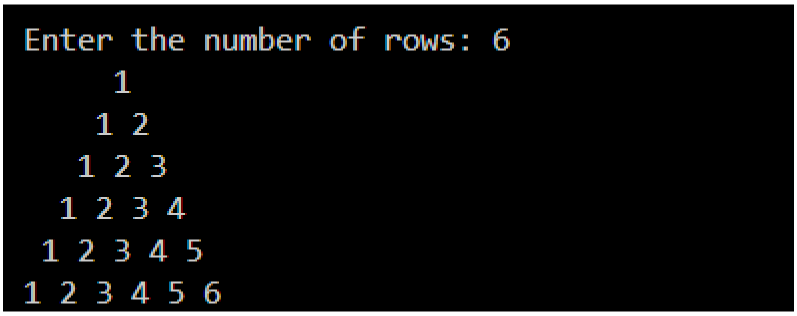
```
bill = calculate_electricity_bill(units)

print("The electricity bill is: Rs.", bill)
```



```
Enter the number of units consumed: 250
The electricity bill is: Rs. 132.5
```

1.3. Print the pyramid of numbers using for loops.

```
rows = int(input("Enter the number of rows: "))


for i in range(1, rows+1):

   for j in range(rows-i):

      print(" ", end="")

   for j in range(1, i+1):

      print(j, end=" ")

   print()
```

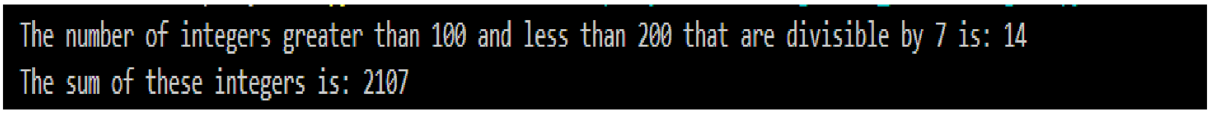


```
Enter the number of rows: 6
     1
    1 2
   1 2 3
  1 2 3 4
 1 2 3 4 5
1 2 3 4 5 6
```

1.4. Write a program to find the number and sum of all integers greater than 100 and less than 200 that are divisible by 7.

```
count = 0
total_sum = 0

for num in range(101, 200):
    if num % 7 == 0:
        count += 1
        total_sum += num

print("The number of integers greater than 100 and less than 200 that are divisible by 7 is:", count)
print("The sum of these integers is:", total_sum)
```

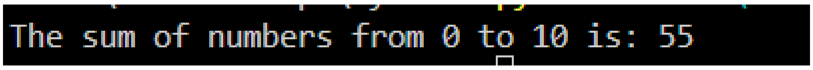
Output



```
The number of integers greater than 100 and less than 200 that are divisible by 7 is: 14
The sum of these integers is: 2107
```

1.5. Write a recursive function to calculate the sum of numbers from 0 to 10

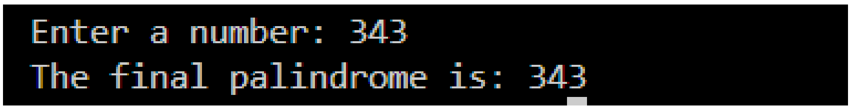
```
def calculate_sum(n):
    if n == 0:
        return 0
    else:
        return n + calculate_sum(n-1)

result = calculate_sum(10)
print("The sum of numbers from 0 to 10 is:", result)
```



```
The sum of numbers from 0 to 10 is: 55
```

1.6. Write a Python program to reverse the digits of a given number and add them to the original. If the sum is not a palindrome, repeat this procedure.

```
def is_palindrome(num):
    original_num = num
    reversed_num = 0

    while num > 0:
        digit = num % 10
        reversed_num = reversed_num * 10 + digit
        num //= 10

    return original_num == reversed_num

def reverse_and_add(num):
    original_num = num
    reversed_num = 0

    while num > 0:
        digit = num % 10
        reversed_num = reversed_num * 10 + digit
        num //= 10

    return original_num + reversed_num

def main():
    num = int(input("Enter a number: "))

    while not is_palindrome(num):
        num = reverse_and_add(num)
    print("The final palindrome is:", num)


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



```
Enter a number: 343
The final palindrome is: 343
```

1.7. Write a menu-driven program that performs the following operations on

strings

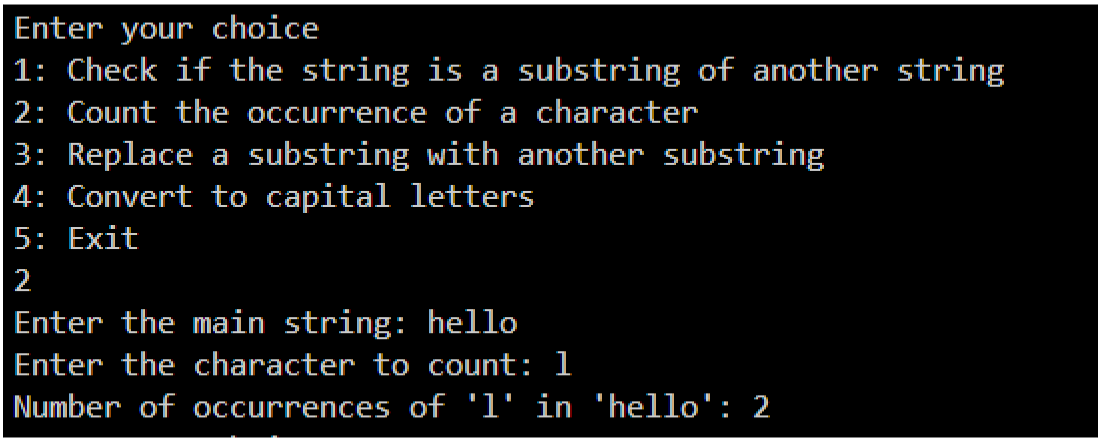
1.7.1. Check if the String is a Substring of Another String

1.7.2. Count Occurrences of Character

1.7.3. Replace a substring with another substring

1.7.4. Convert to Capital Letters

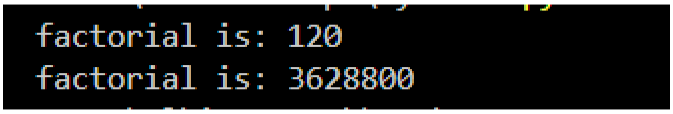
```
n = 0
while n != 5:
    n = int(input("Enter your choice\n1: Check if the string is a substring of another string\n2: Count the occurrence of a character\n3: Replace a substring with another substring\n4: Convert to capital letters\n5: Exit\n"))

    if n == 1:
        main_string = input("Enter the main string: ")
        sub_string = input("Enter the substring to check: ")
        if sub_string in main_string:
            print(f"'{sub_string}' is a substring of '{main_string}'.")
        else:
            print(f"'{sub_string}' is not a substring of '{main_string}'.")

    elif n == 2:
        main_string = input("Enter the main string: ")
        char = input("Enter the character to count: ")
        count = main_string.count(char)
        print(f"Number of occurrences of '{char}' in '{main_string}': {count}")

    elif n == 3:
        main_string = input("Enter the main string: ")
        old_substring = input("Enter the substring to replace: ")
        new_substring = input("Enter the new substring: ")
        modified_string = main_string.replace(old_substring, new_substring)
        print(f"Modified string: '{modified_string}'")

    elif n == 4:
        main_string = input("Enter the string to convert to capital letters: ")
        capital_string = main_string.upper()
        print(f"String in capital letters: '{capital_string}'")

    elif n == 5:
        print("Exiting the program...")

    else:
        print("Invalid choice. Please enter a number from 1 to 5.")

print("Program ended.")
```



```
Enter your choice
1: Check if the string is a substring of another string
2: Count the occurrence of a character
3: Replace a substring with another substring
4: Convert to capital letters
5: Exit
2
Enter the main string: hello
Enter the character to count: l
Number of occurrences of 'l' in 'hello': 2
```

1.8. Write a function to find the factorial of a number but also store the factorials calculated in a dictionary.

```
def factorial(n):
    """
    Calculates the factorial of a number and stores the calculated factorials in a dictionary.
    """
    factorials = {}

    def _factorial(x):
        if x in factorials:
            return factorials[x]
        elif x == 0:
            return 1
        else:
            result = x * _factorial(x-1)
            factorials[x] = result
            return result

    return _factorial(n)

print("factorial is:",factorial(5))
print("factorial is:",factorial(10))
```



```
factorial is: 120
factorial is: 3628800
```

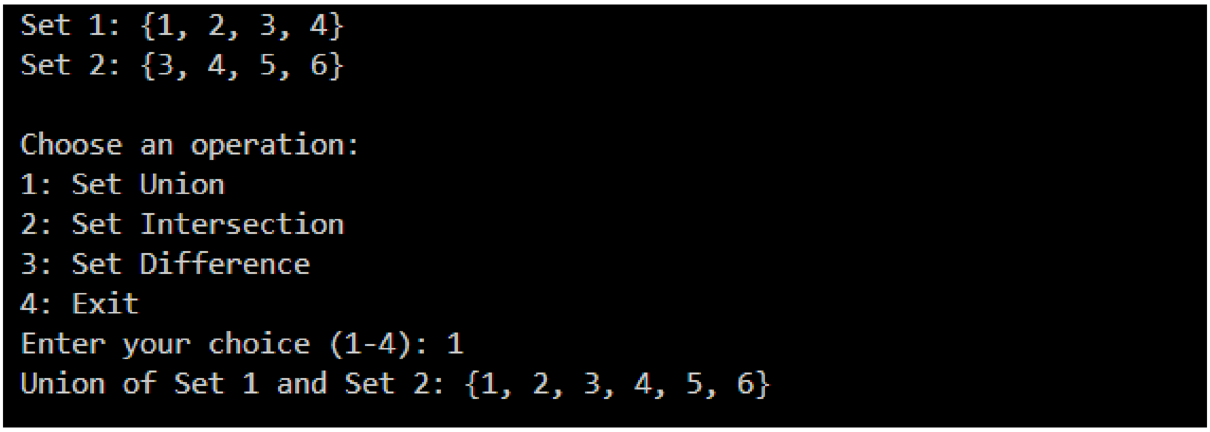
1.9. Perform various set operations

1.9.1. Set Union

1.9.2. Set Intersection

1.9.3. Set Difference

```
def set_union(set1, set2):
    return set1.union(set2)

def set_intersection(set1, set2):
    return set1.intersection(set2)

def set_difference(set1, set2):
    return set1.difference(set2)

# Example usage:
if __name__ == "__main__":
    set1 = {1, 2, 3, 4}
    set2 = {3, 4, 5, 6}

    print("Set 1:", set1)
    print("Set 2:", set2)

    choice = 0
    while choice != 4:
        print("\nChoose an operation:")
        print("1: Set Union")
        print("2: Set Intersection")
        print("3: Set Difference")
        print("4: Exit")

        choice = int(input("Enter your choice (1-4): "))

        if choice == 1:
            print("Union of Set 1 and Set 2:", set_union(set1, set2))
        elif choice == 2:
            print("Intersection of Set 1 and Set 2:", set_intersection(set1, set2))
        elif choice == 3:
            print("Difference (Set 1 - Set 2):", set_difference(set1, set2))
        elif choice == 4:
            print("Exiting the program...")
        else:
            print("Invalid choice. Please enter a number from 1 to 4.")

    print("Program ended.")
```



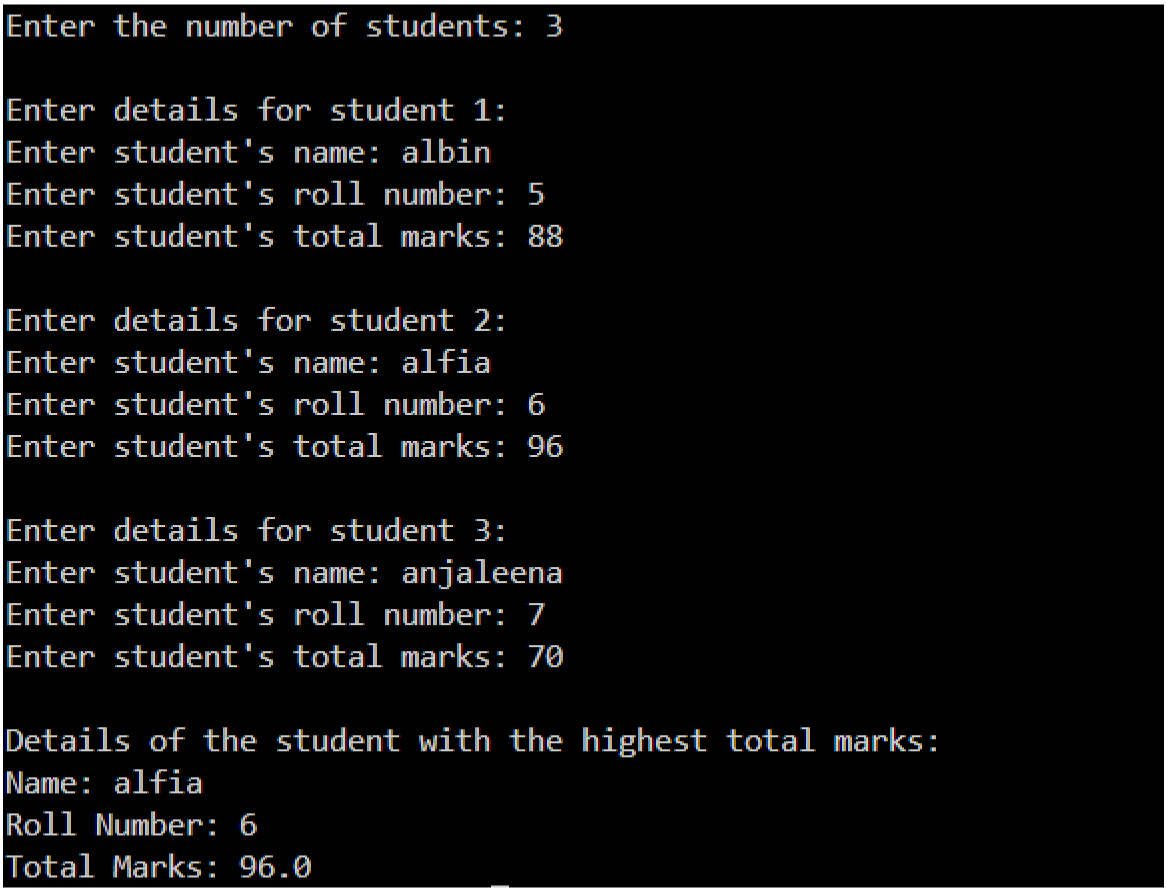
```
Set 1: {1, 2, 3, 4}
Set 2: {3, 4, 5, 6}

Choose an operation:
1: Set Union
2: Set Intersection
3: Set Difference
4: Exit
Enter your choice (1-4): 1
Union of Set 1 and Set 2: {1, 2, 3, 4, 5, 6}
```

1.10. Create a dictionary to store the name, roll_no, and total_mark of N students.

Now print the details of the student with the highest total_mark.

```
def main():
    n = int(input("Enter the number of students: "))

    student_dict = {}

    for i in range(1, n + 1):
        print(f"\nEnter details for student {i}:")
        name = input("Enter student's name: ")
        roll_no = input("Enter student's roll number: ")
        total_marks = float(input("Enter student's total marks: "))

        student_dict[i] = {
            'name': name,
            'roll_no': roll_no,
            'total_marks': total_marks
        }

    highest_marks_student = None
    highest_marks = -1

    for student_id, details in student_dict.items():
        if details['total_marks'] > highest_marks:
            highest_marks = details['total_marks']
            highest_marks_student = details

    if highest_marks_student:
        print("\nDetails of the student with the highest total marks:")
        print(f"Name: {highest_marks_student['name']}")
        print(f"Roll Number: {highest_marks_student['roll_no']}")
        print(f"Total Marks: {highest_marks_student['total_marks']}")
    else:
        print("\nNo student records found.")

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

Output



```
Enter the number of students: 3

Enter details for student 1:
Enter student's name: albin
Enter student's roll number: 5
Enter student's total marks: 88

Enter details for student 2:
Enter student's name: alfia
Enter student's roll number: 6
Enter student's total marks: 96

Enter details for student 3:
Enter student's name: anjaleena
Enter student's roll number: 7
Enter student's total marks: 70

Details of the student with the highest total marks:
Name: alfia
Roll Number: 6
Total Marks: 96.0
```

1.11. Write a Python program to copy the contents of a file into another file, line by line.

1.12. Use the OS module to perform

1.12.1. Create a directory

1.12.2. Directory Listing

1.12.3. Search for “.py” files

1.12.4. Remove a particular file

1.13. Create a simple banking application by using inheritance.