



**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**  
**CHENNAI-602105**

**CSA08 – PYTHON PROGRAMMING**A.Yuvan charan

192424383

1. Write a program to print all the Non-Prime numbers between A and B ?

main.py

Share

Run

Clear

```
1  
2 a = int(input("Enter the value of a: "))  
3 b = int(input("Enter the value of b: "))  
4  
5 print(f"Prime numbers between {a} and {b} are:")  
6  
7  
8 for x in range(a, b + 1):  
9     if x > 1:  
10         for i in range(2, x):  
11             if x % i == 0:  
12                 break  
13         else:  
14             print(x)  
15
```

Output

Enter the value of a: 12  
Enter the value of b: 19  
Prime numbers between 12 and 19 are:  
13  
17  
19  
  
=== Code Execution Successful ===

- 2 Find the year of the given Anniversary is leap year or not. If leap year then print the next Anniversary, if not leap year then print the previous Anniversary.

```
main.py      [Run]      Output      Clear
```

```
1  
2 date = input("Enter the date to be checked (dd/mm/yyyy): ")  
3  
4  
5 c = date.split("/")  
6 b = list(map(int, c))  
7 input_year = b[2]  
8  
9  
10 if (input_year % 4 == 0):  
11     if (input_year % 100 == 0):  
12         if (input_year % 400 == 0):  
13             print(f"Given Anniversary Year: Leap Year.")  
14             print(f"Anniversary Date: {c[0]}/{c[1]}/{input_year + 1}")  
15         else:  
16             print(f"Given Anniversary Year: Non Leap Year.")  
17             print(f"Anniversary Date: {c[0]}/{c[1]}/{input_year - 1}")  
18     else:  
19         print(f"Given Anniversary Year: Leap Year.")  
20         print(f"Anniversary Date: {c[0]}/{c[1]}/{input_year + 1}")  
21 else:  
22     print(f"Given Anniversary Year: Non Leap Year.")  
23     print(f"Anniversary Date: {c[0]}/{c[1]}/{input_year - 1}")  
24  
25
```

```
Enter the date to be checked (dd/mm/yyyy): 04/11/1947  
Given Anniversary Year: Non Leap Year.  
Anniversary Date: 04/11/1946  
  
== Code Execution Successful ==
```

3. Write a program to print the given number is Perfect number or not?

```
main.py  Run  Clear
1 Number=int(input(" Please Enter any Number: "))
2 Sum = 0
3 for i in range(1, Number):
4     if (Number%i==0):
5         Sum=Sum+ i
6 if (Sum==Number):
7     print("%d is a Perfect Number" %Number)
8
9 else:
10    print("%d is not a Perfect Number" %Number)
11
12
```

Output

```
Please Enter any Number: 6
6 is a Perfect Number

=== Code Execution Successful ===
```

4 Write a program to generate Pythagorean Triplets for the given limit.

```
main.py  Run  Clear
1
2 A = input("Enter upper limit: ")
3 c = 0
4 m = 2
5
6 if A.isnumeric():
7     x = int(A)
8     print("Pythagorean Triplets are:")
9     while c < x:
10        for n in range(1, m):
11            a = m * m - n * n
12            b = 2 * m * n
13            c = m * m + n * n
14
15            if c > x:
16                break
17
18            if a == 0 or b == 0 or c == 0:
19                continue
20
21            print(a, b, c)
22            m += 1
23 else:
24     print("Invalid input")
25
```

Output

```
Enter upper limit: 10
Pythagorean Triplets are:
3 4 5
8 6 10

=== Code Execution Successful ===
```

5. Write a program to find the sum of digits of N digit number (sum should be single digit)

```
main.py  Run  Clear
1 # Input from user
2 n = int(input("Enter N value: "))
3 num = int(input(f"Enter {n} digit number: "))
4
5 # Validate the number of digits
6 if len(str(num)) != n:
7     print("Invalid input: Number does not have", n, "digits.")
8 else:
9     # Function to calculate digit sum until it's a single digit
10    def single_digit_sum(number):
11        while number >= 10:
12            temp_sum = 0
13            while number > 0:
14                temp_sum += number % 10
15                number //= 10
16            number = temp_sum
17        return number
18
19    result = single_digit_sum(num)
20    print(f"Sum of {n} digit number: {result}")
21
```

Output

```
Enter N value: 3
Enter 3 digit number: 143
Sum of 3 digit number: 8

=== Code Execution Successful ===
```

## 6. Program to find whether the given number is Armstrong number or not

```
main.py  Run  Output  Clear
1 num = int(input("Enter the number: "))
2 Sum = 0
3 temp = num
4 n = len(str(num))
5
6 while temp > 0:
7     digit = temp % 10
8     Sum += digit ** n
9     temp = temp // 10
10
11 if Sum == num:
12     print("Given number is Armstrong number")
13 else:
14     print("Given number is not an Armstrong number")
15
```

Enter the number: 153  
Given number is Armstrong number  
=== Code Execution Successful ===

## 7. Program to find whether the given number is Harshad number or not

```
main.py  Run  Output  Clear
1 num = int(input("Enter the number: "))
2 Sum = 0
3 temp = num
4
5 while temp > 0:
6     digit = temp % 10
7     Sum += digit
8     temp = temp // 10
9
10 if num % Sum == 0:
11     print("Given number is Harshad number")
12 else:
13     print("Given number is not a Harshad number")
14
```

Enter the number: 21  
Given number is Harshad number  
=== Code Execution Successful ===

## 8. Program to find whether the given number is Happy number or not

```
main.py  Run  Output  Clear
1 def happy(n):
2     sum = 0
3     while n > 0:
4         digit = n % 10
5         sum += digit ** 2
6         n = n // 10
7     return sum
8
9 # Main Program
10 num = int(input("Enter the number: "))
11 result = num
12
13 while result != 1 and result != 4:
14     result = happy(result)
15
16 if result == 1:
17     print("Given number is Happy number.")
18 else:
19     print("Given number is not a Happy number.")
```

Enter the number: 19  
Given number is Happy number.  
=== Code Execution Successful ===

### 9. Program to find whether the given number is Tech number or not

```
main.py  Run  Output  Clear
1 n = int(input("Enter number: "))
2 m = str(n)
3
4 if len(m) % 2 != 0:
5     print("Not a Tech number")
6 else:
7     a = m[:len(m)//2]
8     b = m[len(m)//2:]
9     c = int(a) + int(b)
10    d = c ** 2
11
12    if d == n:
13        print("Given number is Tech number")
14    else:
15        print("Given number is not a Tech number")
```

Enter number: 3025  
Given number is Tech number  
=== Code Execution Successful ===

### 10. Write a program using function to calculate the simple interest. Suppose the customer is a senior citizen. She is being offered 15 percent rate of interest; he is being offered 12 percent rate of interest for all other customers, the ROI is 10 percent.

```
main.py  Run  Output  Clear
1 def calculate_interest(p, n, gender, senior_citizen):
2     if senior_citizen.lower() == 'y':
3         if gender.lower() == 'f':
4             rate = 15
5         else:
6             rate = 12
7     else:
8         rate = 10
9     interest = (p * n * rate) / 100
10    return interest
11
12    p = int(input("Enter the principal amount: "))
13    n = int(input("Enter the number of years: "))
14    gender = input("Gender (m/f): ")
15    senior = input("Is customer senior citizen (y/n): ")
16
17    interest = calculate_interest(p, n, gender, senior)
18    print("Interest:", int(interest))
```

Enter the principal amount: 200000  
Enter the number of years: 3  
Gender (m/f): m  
Is customer senior citizen (y/n): n  
Interest: 60000  
=== Code Execution Successful ===

### 11. Find the number of factors for the given number and print the 1st N factors of the given number.

```
main.py  Run  Output  Clear
1 x = int(input("Enter the number: "))
2 n = int(input("Enter N value: "))
3
4 factors = []
5
6 for i in range(1, x + 1):
7     if x % i == 0:
8         factors.append(i)
9
10    print("Number of factors =", len(factors))
11
12    if n > len(factors):
13        print("Invalid: N is greater than the number of factors")
14    else:
15        print(f"1st {n} factors are:", end=" ")
16        for i in range(n):
17            print(factors[i], end=" " if i != n - 1 else "")
```

Enter the number: 100  
Enter N value: 4  
Number of factors = 9  
1st 4 factors are: 1, 2, 4, 5  
=== Code Execution Successful ===

12. Write a program to print number of factors and to print nth factor of the given number.

```
main.py  Run  Output  Clear
1 x = int(input("Enter the number: "))
2 factors = []
3 for i in range(1, x + 1):
4     if x % i == 0:
5         factors.append(i)
6 print("Number of factors =", len(factors))
7 n = int(input("Enter N value: "))
8 if 1 <= n <= len(factors):
9     print(f"{n}th factor of {x} =", factors[n - 1])
10 else:
11     print("Invalid: N is out of range")
```

Enter the number: 100  
Number of factors = 9  
Enter N value: 4  
4th factor of 100 = 5  
=== Code Execution Successful ===

13. Write a program to print unique permutations of a given number Sample.

```
main.py  Run  Output  Clear
1 import itertools
2 n = input("Enter the number: ")
3 P = set(itertools.permutations(n))
4 print("Permutations are:")
5 for p in sorted(P):
6     print("".join(p))
```

Enter the number: 143  
Permutations are:  
134  
143  
314  
341  
413  
431  
=== Code Execution Successful ===

14. Write a program to find the square, cube of the given decimal number

```
main.py  Run  Output  Clear
1 import math
2 num = float(input("Enter the number: "))
3 print("Square number =", round(math.pow(num, 2), 3))
4 print("Cube number =", round(math.pow(num, 3), 3))
```

Enter the number: 0.6  
Square number = 0.36  
Cube number = 0.216  
=== Code Execution Successful ===

15. Write a program to convert the Binary to Decimal, Octal Sample Input:

main.py		Output
<pre>1 num = input("Enter the binary number: ") 2 bin_num = "01" 3 binary = True 4 5 for i in range(len(num)): 6     if num[i] not in bin_num: 7         print("Invalid input") 8         binary = False 9         break 10 11 if binary: 12     dec_number = int(num, 2) 13     oct_number = oct(dec_number)[2:] 14 15     print("Decimal Equivalent =", dec_number) 16     print("Octal Equivalent =", oct_number) 17</pre>		<pre>Enter the binary number: 1101 Decimal Equivalent = 13 Octal Equivalent = 15  === Code Execution Successful ===</pre>