01. Develop a python program to find the better of two test average marks out of three tests mareks accepted from the user

```python
mark1 = float(input("Enter mark for Test 1: "))
mark2 = float(input("Enter mark for Test 2: "))
mark3 = float(input("Enter mark for Test 3: "))


marks = [mark1, mark2, mark3]


marks.sort(reverse=True)


best_two = marks[:2]


average = sum(best_two) / 2


print("The better average of the two highest marks is:", average)
```

```
Enter mark for Test 1: 78
Enter mark for Test 2: 89
Enter mark for Test 3: 56
The better average of the two highest marks is: 83.5
```

02. Develop a python program to find the smallest and largest number in a list

```python
numbers = input("Enter numbers separated by spaces: ")


num_list = list(map(float, numbers.split()))


smallest = min(num_list)
largest = max(num_list)


print("Smallest number is:", smallest)
print("Largest number is:", largest)
```

```
Enter numbers separated by spaces: 4 7 1 9 3
Smallest number is: 1.0
Largest number is: 9.0
```

03.Develop a python program to arrange the numbers in ascending and descending order

```python
numbers=input("Enter numbers seperated by space: ")

num_list=list(map(int,numbers.split()))

ascending=sorted(num_list)

descending=sorted(num_list,reverse=True)

print("Ascending order:",ascending)
print("Descending order:",descending)
Enter numbers seperated by space: 45 12 65 32 2 8
Ascending order: [2, 8, 12, 32, 45, 65]
Descending order: [65, 45, 32, 12, 8, 2]
```

04 Develop a binary search program in python

```python
def binary_search(arr, target):
    start = 0
    end = len(arr) - 1

    while start <= end:
        mid = (start + end) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            start = mid + 1
        else:
            end = mid - 1
    return -1


numbers = [2, 4, 6, 8, 10, 12]
target_number = 8

index = binary_search(numbers, target_number)

if index != -1:
    print(f"Found {target_number} at index {index}")
else:
    print(f"{target_number} not found in the list")

Found 8 at index 3
```

05 Develop a bubble sort program in python

```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):

        for j in range(0, n - i - 1):

            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]



numbers = [64, 34, 25, 12, 22, 11, 90]
bubble_sort(numbers)
print("Sorted array is:", numbers)
```

```
Sorted array is: [11, 12, 22, 25, 34, 64, 90]
```

06 Develop a Python program to check whether a given number is palindrome or not and also count the

```python
def check_palindrome_and_count(number):

    num_str = str(number)


    if num_str == num_str[::-1]:
        print(f"{number} is a palindrome.")
    else:
        print(f"{number} is not a palindrome.")


    digit_count = len(num_str)
    print(f"Number of digits: {digit_count}")


num = int(input("Enter a number: "))
check_palindrome_and_count(num)
```

```
Enter a number: 55
55 is a palindrome.
Number of digits: 2
```

07 Write a Python program that accepts a sentence and find the number of words, digits, Uppercase letters and lowercase letters

```python
def analyze_sentence(sentence):
    word_count = len(sentence.split())
```

```python
    digit_count = 0
    uppercase_count = 0
    lowercase_count = 0

    for char in sentence:
        if char.isdigit():
            digit_count += 1
        elif char.isupper():
            uppercase_count += 1
        elif char.islower():
            lowercase_count += 1

    print("Number of words:", word_count)
    print("Number of digits:", digit_count)
    print("Number of uppercase letters:", uppercase_count)
    print("Number of lowercase letters:", lowercase_count)


text = input("Enter a sentence: ")
analyze_sentence(text)
```

```
Enter a sentence: Hello world 123
Number of words: 3
Number of digits: 3
Number of uppercase letters: 1
Number of lowercase letters: 9
```

08 Write a Python program for pattern recognition with and without using regular expressions

```python
def is_phone_number(text):
    if len(text) != 12:
        return False
    if text[3] != '-' or text[7] != '-':
        return False
    if not (text[:3].isdigit() and text[4:7].isdigit() and text[8:].isdigit()):
        return False
    return True


sample = input("Enter a string: ")
if is_phone_number(sample):
    print("Valid phone number pattern found!")
else:
    print("No valid phone number pattern found.")
```

```
Enter a string: 123-456-7890
```

Valid phone number pattern found!