

## **PYTHON ASSIGNMENT**

**NAME** : Moideen Nihal

**REGISTER NUMBER** : PROV/BCA/7/24/059

### **Program -1**

```
a=int(input('Enter the first number: '))
b=int(input('Enter the second number: '))
adding=a+b
subtraction=a-b
multiplication=a*b
division=a/b
modulus=a%b
exponential=a**2
floor=a//b
print('adding: ', adding)
print('subtraction: ', subtraction)
print('multiplication: ', multiplication)
print('division: ', division)
print('modulus: ', modulus)
print('exponential: ', exponential)
print('floor:', floor)
```

### **Output:**

```
a=60
b=10
adding: 70
subtraction: 50
multiplication: 600
division: 6.0
modulus: 0
exponential: 6046617600000000000
floor: 6
```

**Explanation :** This program uses arithmetic operators such as +, -, \*, /, %, \*\*, and // to take two values, a and b, and displays the results with the corresponding operations.

### Program-2

```
a=int(input('Enter first number: '))
b=int(input('Enter second number: '))

if a>b:
    print('first number is greater than second')
elif a==b:
    print('first number is equal to second number')
elif a<=b:
```

#### output:

```
first number: 23
enter second number: 12
first number is greater than second number
```

**Explanation:** This program utilizes comparison operators (such as > and =) and employs if-elif-else conditional statements to help determine the greater and lesser of two numbers.

### Program-3

```
a=input("enter a boolean value 1:-").strip().lower()=="true"
b=input("enter a boolean value 2:-").strip().lower()=="true"
c=input("enter a boolean value 3:-").strip().lower()=="true"

print(a and b and c)#
print(a or b or c)
print(not a)
print(not b)
print(not c)
```

#### Output:

```
True
False
True
False
True
False
```

True

False

**Explanation:** This program employs logic gates: AND, OR, and NOT. The AND gate outputs true only when both values are true; otherwise, it outputs false. The OR gate outputs true if at least one value is true. The NOT gate inverts the input, producing the opposite value and requiring only one input.

#### Program-04

```
a=input('Enter a word: ')
b=len(a)
c=a[0],a[-1]
d=a[::-1]
e=a.upper()
f=a.lower()
print(b)
print(c)
print(d)
print(e)
print(f)
```

**output:**

```
6
('N', 'l')
lahiN
NIHAL
nihal
```

**Explanation :** In this string program, we find the length of a string using the length function, which counts spaces as well. We use index positions to identify the first and last letters in the string. To convert a word to uppercase, we use the `upper()` method, and to convert it to lowercase, we use the `lower()` method.

#### Program-5

```
a=input('Enter a name: ')
b=int(input('Enter the age: '))
print('Hello',a,'you are',b,'years old')
```

**Output:**

enter your name= Nihal

enter your age= 18

Hello Nihal you are 18 years old

**Explanation:** This is a simple program. We prompt the user for their name and age, then use the variables to display a message like, "Hello, [name], you are [age] years old." It's easy to execute!

### Program-6

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

```
b=input("Enter the word:")
```

```
c=a.find(b)
```

```
if c !=-1:
```

```
    print(f"The word {b} found at {c} index position")
```

```
else:
```

```
    print(f"The {b} word not found")
```

### Output:

Enter a sentence: Nihal is great

Enter the word: great

The word great found at 11 index position

**Explanation:** In this program, we find the index position using ``print(r.index(s))``. We also use if-else statements to handle conditions where the specified value may not be present. If the condition is not met, the program will execute the alternative statement.

### Program-7

```
a=[]
```

```
for i in range(1,6):
```

```
    b=int(input('enter a number: '))
```

```
    a.append(b)
```

```
c=sum(a)
```

```
d=max(a)
```

```
e=min(a)
```

```
print(a)
```

```
print(c)
```

```
print(d)
```

```
print(e)
```

**output:**

```
[45, 46, 48, 49, 50]
```

```
238
```

```
50
```

```
45
```

**Explanation:** This program utilizes the `sum()` function to calculate the total and employs the `max()` and `min()` functions to identify the largest and smallest values, respectively.

### Program-8

```
fruits=['apple', 'mango', 'grape', 'orange', 'watermelon']
```

```
print(fruits)
```

```
fruits.append('pineapple')
```

```
print(fruits)
```

```
fruits.remove('grape')
```

```
print(fruits)
```

**output:**

```
['apple', 'mango', 'grape', 'orange', 'watermelon']
```

```
['apple', 'mango', 'grape', 'orange', 'watermelon', 'pineapple']
```

```
['apple', 'mango', 'orange', 'watermelon', 'pineapple']
```

**Explanation:** In list manipulation, we use the `append()` function to add new items to the end of the list. To remove elements, we use the `pop()` function, specifying the index of the item to be removed. Finally, we can print the updated list using a print statement.

### Program-9

```
A = int(input("Enter the number 1: "))
```

```
B = int(input("Enter the number 2: "))
```

```
C = int(input("Enter the number 3: "))
```

```
D = int(input("Enter the number 4: "))
```

```
E = int(input("Enter the number 5: "))
```

```
f = [A, B, C, D, E]
```

```
f.sort()
```

```
print("Ascending Order of the list: ", f)
```

```
f.sort(reverse=True)
```

```
print("Descending Order of the list: ", f)
```

**output:** Enter the number 1: 23

Enter the number 2: 6

Enter the number 3: 90

Enter the number 4: 76

Enter the number 5: 0

Ascending Order of the list: [0, 6, 23, 76, 90]

Descending Order of the list: [90, 76, 23, 6, 0]

**Explanation:** This program is designed for sorting a list. First, we take input from the user and then sort it accordingly. To sort in ascending order, we set `reverse=True`, and for descending order, we set `reverse=False`.

### Program -10

```
Numbers=[1,2,3,4,5,6,7,8,9,10]
```

```
print(Numbers[:5])
```

```
print(Numbers[-5:])
```

```
print(Numbers[1:7])
```

**output:** [1, 2, 3, 4, 5] [6, 7, 8, 9, 10] [2, 3, 4, 5, 6, 7]

**Explanation:** The format used here is `[starting:ending]`. In this program, we use indexes to locate elements in the list. For `[:5]`, it retrieves elements from the start up to the 4th element (index 0 to 4). The `[-5:]` notation starts from the 5th element from the end and goes to the last element. Meanwhile, `[1:5]` starts from the 1st element and goes up to the 4th element (index 1 to 4).