

# INTORDUCTION TO ALGORITHMS – EC351

## ASSIGNMENT – 2

### 1.Find the sum of two numbers A and B

**Sol :**

```
from datetime import datetime
start_time = datetime.now()
X = float(input('Enter the value of X : '))
Y = float(input('Enter the value of Y : '))
Z= X+Y
print('The value of Z : ',Z)
end_time = datetime.now()
print('Duration:{ }' .format(end_time - start_time))
```

#### Algorithm

Step-1 Start

Step-2 Input first number say A O(1)

Step-3 Input second number say B O(1)

Step-4 SUM = A + B O(1)

Step-5 Display SUM O(1)

Step-6 Stop

#### Observations

Time complexity =  $O(1) + O(1) + O(1) + O(1) = O(1)$  . Its time complexity is constant in time.

### 2. Convert temperature from Celsius(C) to Fahrenheit (F) and Fahrenheit to Celsius

**Sol : a) Celsius(C) to Fahrenheit (F)**

```

from datetime import datetime
start_time = datetime.now()
celsius=float(input('Enter the celsius : '))
fahrenheit = (celsius * 1.8) + 32
print(celsius,' celsius is equal to',fahrenheit)
end_time = datetime.now()
print('Duration : {}'.format(end_time - start_time))

```

### Algorithm

Step-1. Start

Step-2 Input temperature in Celsius say C                      O(1)

Step-3  $F = (\text{celsius} * 1.8) + 32$                       O(1)

Step-4 Display Temperature in Fahrenheit F,C              O(1)

Step-5 Stop

### Observations

Time complexity =  $O(1) + O(1) + O(1) = O(1)$ . Time complexity is constant time.

## B) Fahrenheit to Celsius

```

from datetime import datetime
start_time = datetime.now()
Fahrenheit=float(input('Enter the Fahrenheit : '))
celsius = (Fahrenheit - 32) / 1.8
print(Fahrenheit,' Fahrenheit is equal to',celsius)
end_time = datetime.now()
print('Duration : {}'.format(end_time - start_time))

```

### Algorithm

Step-1. Start

Step-2 Input temperature in Celsius say C                      O(1)

Step-3  $F = (\text{Fahrenheit} - 32) / 1.8$                       O(1)

Step-4 Display Temperature in celsius F,C       $O(1)$

Step-5 Stop

### Observations

Time complexity =  $O(1) + O(1) + O(1) = O(1)$ . Time complexity is constant time.

### 3.Find Area(A) and Perimeter (P)of a Square

**Sol :**

```
from datetime import datetime
start_time = datetime.now()
X=float(input('Enter the side : '))
Perimeter = 4*X
Area = X*X
print('Perimeter of Square : ',Perimeter)
print('Area of Square : ',Area)
end_time = datetime.now()
print('Duration : {}'.format(end_time - start_time))
```

### Algorithm

Step-1 Start

Step-2 Input Side Length of Square say X       $O(1)$

Step-3 Area = X x X       $O(1)$

Step-4 PERIMETER = 4 x X       $O(1)$

Step-5 Display AREA, PERIMETER       $O(1)$

Step-6 Stop

### Observations

Time complexity =  $O(1) + O(1) + O(1) + O(1) = O(1)$ . Time complexity is constant time.

### 4. Find the Compound Interest (CI)

**Sol :**

```

from datetime import datetime
start_time = datetime.now()
p = float(input("Enter the principle amount : "))
r = float(input("Enter the rate of interest : "))
t = float(input("Enter the time in the years: "))
Compound_Intrest = p * (pow((1 + r / 100), t))
print("Principle amount : ", p)
print("Interest rate    : ", r)
print("Time in years    : ", t)
print("compound Interest : ", Compound_Intrest)
end_time = datetime.now()
print('Duration : {}'.format(end_time - start_time))

```

### Algorithm

Step-1 Start

Step-2 Input value of p, t, r O(1)

Step-3  $\text{Compound\_Intrest} = p * (\text{pow}((1 + r / 100), t))$  O(1)

Step-4 Display CI O(1)

Step-5 Stop

### Observations

Time complexity =  $O(1) + O(1) + O(1) = O(1)$ . Time complexity is constant time.

## 5. Swap Two Numbers using Temporary Variable

**Sol :**

```

from datetime import datetime
start_time = datetime.now()
x = float(input("Enter the Value of X : "))
y = float(input("Enter the Value of Y : "))

```

```

temp = x
x = y
y = temp
print('The value of x after swapping: ',x)
print('The value of y after swapping: ',y)
end_time = datetime.now()
print('Duration : {}'.format(end_time - start_time))

```

### **Algorithm**

Step-1 Start

Step-2 Input Two Numbers Say x, y O(1)

Step-3 TEMP = x O(1)

Step-4 x = y O(1)

Step-5 y = temp O(1)

Step-6 Display After Swap Values x, y O(1)

Step-7 Stop

### **Observations**

Time complexity =  $O(1)+O(1)+O(1)+O(1)+O(1) = O(1)$ . Time complexity is constant time.

## **6. Find the Smallest of two numbers A and B**

**Sol :**

```

from datetime import datetime
start_time = datetime.now()
x=float(input('Enter the value of X : '))
y=float(input('Enter the value of Y : '))
if x<y:
    print('The smallest number is : ',x)
else:

```

```
print("The smallest number is : ',y)
end_time = datetime.now()
print('Duration : {}'.format(end_time - start_time))
```

### **Algorithm :**

Step-1 Start

Step-2 Input two numbers say x, y

Step-3 If  $x < y$  print smallest is x O(1)

Else print smallest is y O(1)

Step-4 Stop

### **Observations :**

Time complexity =  $O(1) + O(1) = O(1)$ . Time complexity is constant time.

## **7. Find the largest of three numbers A, B and C**

### **Sol :**

```
from datetime import datetime
start_time = datetime.now()
x = float(input("Enter first number: "))
y = float(input("Enter second number: "))
z = float(input("Enter third number: "))
if (x >= y) and (x >= z):
    largest = x
elif (y >= x) and (y >= z):
    largest = y
else:
    largest = z
print("The largest number is", largest)
end_time = datetime.now()
print('Duration : {}'.format(end_time - start_time))
```

## Algorithm

Step-1 Start

Step-2 Input three numbers say x, y, z

Step-3 if (x >= y) and (x >= z): O(1)

largest = x

elif (y >= x) and (y >= z): O(1)

largest = y

else: O(1)

largest = z

Step-4 Stop

## Observations

Time complexity =  $O(1) + O(1) + O(1) = O(1)$ . Time complexity is constant time.

## 8. Find Even number between 1 to 50

**Sol :**

```
from datetime import datetime
```

```
start_time = datetime.now()
```

```
for num in range(1,50):
```

```
    if num%2==0:
```

```
        print ('The even numbers are : ',num)
```

```
    num=num+1
```

```
end_time = datetime.now()
```

```
print('Duration : {}'.format(end_time - start_time))
```

## Algorithm :

Step-1 Start

Step-2 num = 1

Step-3 if num%2==0 O(1)

Step-4 Display num O(1)

Step-5 num=num+1 O(1)

Step-6 Stop

### Observations

Time complexity =  $O(1) + O(1) + O(1) = O(1)$ . Time complexity is constant time

## 9. Find Sum of Series 1+2+3+.....+N

**Sol :**

```
from datetime import datetime
start_time = datetime.now()
num = float(input('Enetr the Number : '))
if num < 0:
    print("Enter a positive number")
else:
    sum = 0
    while(num > 0):
        sum += num
        num -= 1
    print("The sum is", sum)
end_time = datetime.now()
print('Duration : {}'.format(end_time - start_time))
```

### Algorithm :

Step-1 Start

Step-2 Input Value of Num

Step-3 if num < 0 O(1)

Step-4 Display that to enter a positive num O(1)

Step-5 else sum = 0 O(1)



Step-6 while(num > 0) O(n)

    sum += num

    num -= 1

Step-7 Display the sum O(1)

Step-8 Stop

### **Observations**

Time complexity =  $O(1) + (1) + O(1) + O(n) + O(1) = O(n)$ .

Time complexity is linear time here because the time complexity depends on the value of 'n' and number of times the loop repeats .

by

Y Santhi Swarup

