Program No.: 02

**Aim:**

Implementation of basic concepts of variables, operators, value types, statements, conditional statements and loops.

**Topics covered:**

Variables, operators, value types, statements, conditional statements and loops.

**Course Outcome**

CO1: Implement the basic concepts of Python.

**Case Studies:**

***Case Study 01:***

***Problem Statement:***

Python script to convert a given number(in days) to years, months and days, assumes that all months have 30 days and a year have 365 days.

***Problem Solution:***

1. Take a number of days as input.

2. To find number of years divide days with 365.

3. To find number of months divide remaining days by 30.

4. The remaining numbers gives number of days.

5. Exit.

***Program/Source Code:***

#Practice program to convert given days into years months and days.

"""

Case study : 01

File name : cse1.py

Topics : Variables, operators, value types and conditional statements

"""

# read an input of number from the user

num\_of\_days=int(input("Enter the number of days:"))

# converting into years,months and days

year=int((num\_of\_days)/365)

month=int((num\_of\_days%365)/30)

day=int((num\_of\_days%365)%30)

#print years

print("after conversion")

print "years:",year

#print months

print "months:",month

#print days

print "days:",day

***Program Explanation:***

1. Take a number of days as input and store it in the variable num\_of\_days.

2. Find number of years by dividing(/) the days by 365 and store it in the variable year.

3. Store the remainder days in the variable num\_of\_days.

4. Find number of months by dividing the remaining days by 30 and store it in the variable months.

5. Store the remainder days in the variable num\_of\_days.

6. The remainder is the number of days.

7. Print the output I.e., the no. of years, months and days.

8.Exit.

***Runtime Test Cases:***

1. Enter the number of days:700

after conversion

years: 1

months: 11

days: 5

1. Enter the number of days:700

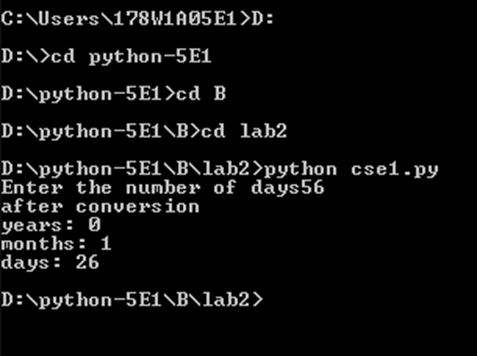
after conversion

years: 5

months: 5

days: 25

**Output:**



**Result:**

Python script to convert a given number (in days) to years, months and days done successfully.

***Case Study 02:***

***Problem Statement:***

Develop a python program that reads a temperature value in Celsius degree from the keyword and transforms it to Fahrenheit degrees. The program must print the two values in the form: X Celsius degrees are Y Fahrenheit degrees.

***Problem Solution:***

1. Take a number of celsius as input.

2. Use the formula T(F)=(T(C)\*1.8)+32.

3. Print T(F)

4. Exit.

***Program/Source Code:***

#Practice program to convert temperature in Celsius to Fahrenheit.

"""

Case study : 02

File name : cse3.py

Topics : Variables, operators, value types and conditional statements

"""

# read an input of number from the user

num\_of\_cel=int(input("Enter the number of celsius:"))

# converting celsius into fahrenheit

fahren=int(((num\_of\_cel)\*1.8)+32)

#print output

print("after conversion")

print "fahren:",fahren,”F”

***Program Explanation:***

1. Take a number as input and store it in the variable num\_of\_cel.

2. Use the formula T(F)=(T(C)\*1.8)+32 to convert the given temperature into Fahrenheit.

3. Store the value we got in the variable fahren.

4. Print the values stored in fahrenheit.

5.Exit.

***Runtime Test Cases:***

1. Enter the number of celsius:23.5

after conversion

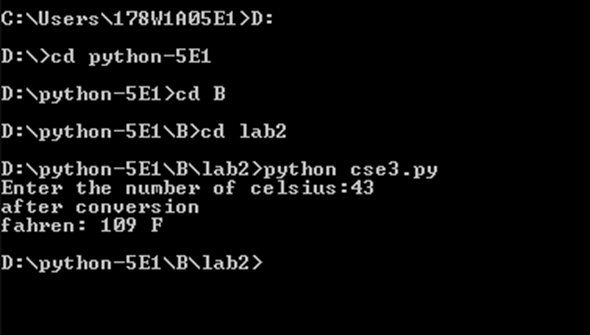
fahren: 74.3F

1. Enter the number of celsius:15.33

after conversion

fahren=59.594F

**Output:**



**Result:**

Python program that reads a temperature value in Celsius degrees from the keyboard and transforms it to Fahrenheit degrees is successfully done.

***Case Study 03:***

***Problem Statement:***

Develop a python program that reads three coefficients (a,b,c) of a 2ndgrade polynomial (ax2+bx+c=0) and obtains and prints the values of x.

***Problem Solution:***

1. Take the inputs of the coefficients a,b and c.

2. Calculate the discriminant and then the roots using the formula.

3. Display the roots.

4. Exit.

***Program/Source Code:***

#Practice scripts to obtain the roots of 2nd degree polynomial.

"""

Case study : 03

File name : cse5.py

Topics : Variables, operators, values and conditional statements

"""

# read an input of number from the user

a=float(input(“Enter the a value:”))

b=float(input(“Enter the b value:”))

c=float(input(“Enter the c value:”))

#logic

x=float(-b+math.sqrt(b\*\*2-4\*a\*c)/2\*a)

y=float(-b-math.sqrt(b\*\*2-4\*a\*c)/2\*a)

#print output

print "sol1:",x

print "sol2:",y

***Program Explanation:***

1. Take the input of each coefficient of the quadratic equation and store it variables a, b, c

respectively in decreasing order.

2. Calculate the discriminant of the equation as shown and store the result in variable dis.

3. Let x and y be the required roots.

4. Calculate x and y by the above formula.

5. Print the resultant x and y values.

6. Exit.

***Runtime Test Cases:***

1. Enter the a value: 2

Enter the b value:- 3

Enter the c value: 1

sol1: 1.0

sol2: 0.5

1. Enter the a value: 2

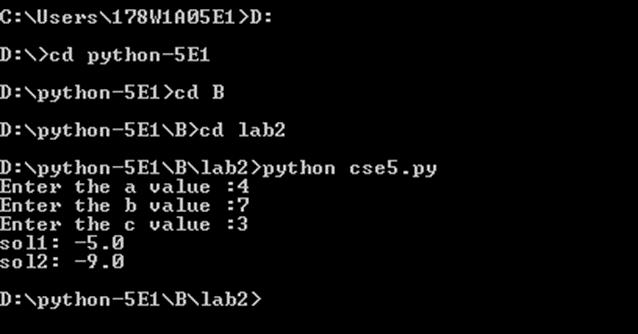
Enter the b value: -5

Enter the c value: 3

sol1: 1.5

Sol2: 1.0

**Output:**



**Result:**

Python script to find roots of a quadratic is successfully done.

***Case Study 04:***

***Problem Statement:***

A Python program to check whether a given number (accept from user) is even or odd and print out an appropriate message to the user.

***Problem Solution:***

1. Take the inputs from the user.

2. Check whether the given number is even or odd.

3. Display the result.

4. Exit.

***Program/Source Code:***

#Practice scripts to check whether the given number is even or odd

"""

Case study : 04

File name : cse4.py

Topics : Variables, conditional statements

"""

# read an input of number from the user

a=int(input(“Enter the a value:”))

#even or odd

If(a%2==0):

print a,”is even number”

else:

print a,”is odd number”

***Program Explanation:***

1. Take the input number from the user and store it in the variable num.

2. Now we check if the number is divisible by 2 by using the modulo operator(%).

3. If it is divisible by 2(i.e. the if condition is true), then display “The given number is even”.

4. Else display “The given number is odd”.

5. Exit.

***Runtime Test Cases:***

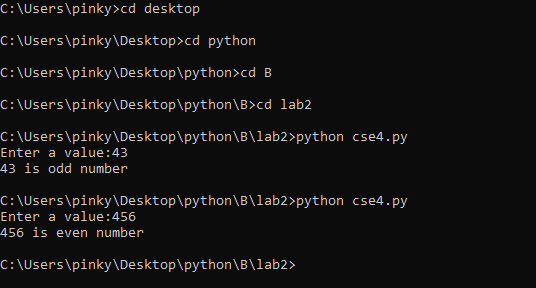
1. Enter the Number: 10

The given number is even.

2. Enter the Number: 53

The given number is odd.

**Output:**



**Result:**

The python script to check whether a given number is even or odd using conditional statements is successfully done

***Case Study 05:***

***Problem Statement:***

A Python program to calculate the factorial of a number using for loop.

***Problem Solution:***

1. Take the inputs from the user.

2. Start from 1 and multiply the product with every number less than or equal to the given number and store the result.

3. Display the result.

4. Exit.

***Program/Source Code:***

#Practice scripts to find factorial of a given number

"""

Case study : 05

File name : cse5.py

Topics : Looping statements

"""

num = int(input("Enter the Number:"))

f = 1

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

f=f \* i

print("Factorial :",f)

***Program Explanation:***

1. Take the input number from the user and store it in the variable num.

2. Initiate a variable num1 to 1.

3. Now in the for loop, start from 1 and multiply the value in num1 with the value.

4. Store the result in num1 again.

5. Repeat steps 2 and 3 until the value reaches num.

6. Display the factorial value:

7. Exit

***Runtime Test Cases:***

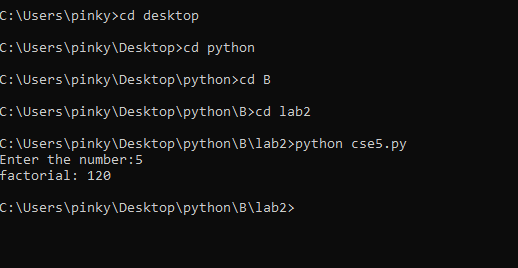
1. Enter the Number: 5

Factorial : 120

2. Enter the Number: 6

Factorial : 720

**Output:**



**Result:**

The python script to calculate factorial of a given number using for loop is successfully done.

***Case Study 06:***

***Problem Statement:***

A python program to check whether number is Palindrome or not.

***Problem Solution:***

1. Take the input number from the user.

2. Reverse the number and store it.

3. Compare the reversed number with the actual number.

4. If they are the same then the number is “palindrome” else it is not.

5. Exit.

***Program/Source Code:***

#Practice scripts to check whether a given number is palindrome or not

"""

Case study : 06

File name : cse6.py

Topics : Conditional and Looping statements.(while)

"""

num = int(input("Enter the Number:"))

n1 = num

s = 0

while(num > 0):

d = num % 10

s = s \* 10 + d

num = num // 10

if (s == n1):

print("Given number is palindrome")

else:

print("Given number is not a palindrome")

***Program Explanation:***

1. Take the input number from the user and store it in the variable num.

2. Initiate a variable num1 to num.

3. Now start reversing the number by accessing its digits from the end.

4. Multiply sum with remainder and repeat the process until no more digits are left in n.

5. Check whether the given number is equal to its reverse. If true print “Given number is palindrome”

6. Else print “Given number is not palindrome”.

7. Exit.

***Runtime Test Cases:***

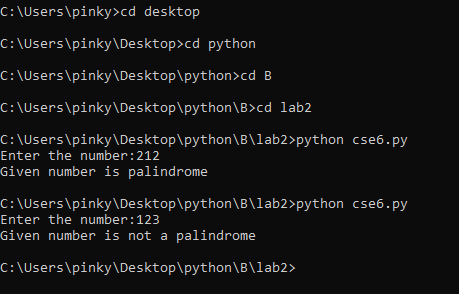
1. Enter the Number:121

Given number is palindrome

2. Enter the Number:12345

Given number is not a palindrome

**Output:**



**Result:**

The python script to check whether a given number is palindrome or not is successfully completed.

***Case Study 07:***

***Problem Statement:***

Write a python program to check whether number is prime or not with while and if statements.

***Problem Solution:***

1. Take the input number from the user.

2. Check whether the given number is divisible by any number other than 1 and itself.

3. If yes then display that it is not a prime.

4. Else display it is prime.

5. Exit.

***Program/Source Code:***

#Python script to check whether a given number is prime or not

"""

Case study : 07

File name : cse7.py

Topics : Conditional suites, Iteration statements.

"""

opt=int(input("Enter the Option 1.to check prime number 2.Display primes in b/w range"))

if(opt==1):

n=int(input("Enter the Number :"))

f=True

i=2

while(i<= n/2):

if(n%i==0):

f=False

i += 1

if(f):

print n," is Prime"

else:

print n," is not prime"

elif(opt==2):

s=0

m=int(input("Enter start range"))

n=int(input("Enter end range"))

for i in range(m,n+1):

f=True

for j in range(2,i/2):

if(i%j==0):

f=False

if(f):

s+=i

print("Sum of Primes in the given range :",s)

else:

print("Invalid Option")

***Program Explanation:***

1. Take the input number from the user and store it in the variable num.

2. Initiate i to 1 and start checking if i divides the number.

3. If the i divides then increment the c value. Later increment i.

4. If I reaches to the given number, break from the while loop.

5. Check if the number c is 2. IF yes then print the given number is prime.

6. Else print “Given number is not prime”.

7. Exit.

***Runtime Test Cases:***

1. Enter the number:9

9 is not prime.

1. Enter the number:11

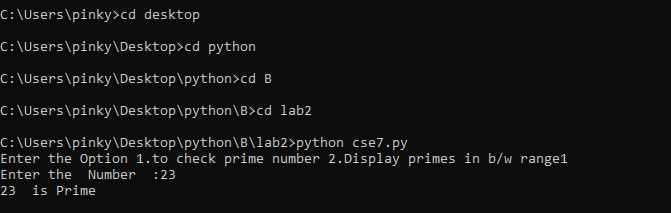
11 is prime.

3. Enter start range:1

Enter end range:10

Sum of Primes in the given range :17

**Output:**



**Result:**

The python script to check whether a given number is prime or not is successfully done.

***Case Study 08:***

***Problem Statement:***

A python program to print all Leap Year from 1 to N.

***Problem Solution:***

1. Take a number as input.

2. Use for loop to iterate the numbers from 1 to number using iterator i.

2.1 If i is divisible by 400, print i.

2.2 Else if i is divisible by 4 and not divisible by 100, print i.

3. Exit.

***Program/Source Code:***

#Python program to print leap year 1 to n

"""

Case study : 08

File name : cse8.py

Topics : Conditional and Looping statements

"""

#Taking input from user

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

#Logic

for i in range(1,num) :

if(i%400 == 0) :

print(i)

elif (i%4==0 and i%100!=0) :

print(i)

***Program Explanation:***

1. Take a number as input and store it in the variable num.

2. Use for loop to iterate from 1 to n using iterator i.

2.1 If i is divisible by 400, print i.

2.2 Else if i is divisible by 4 and not divisible by 100, print i.

3. Exit.

***Runtime Test Cases:***

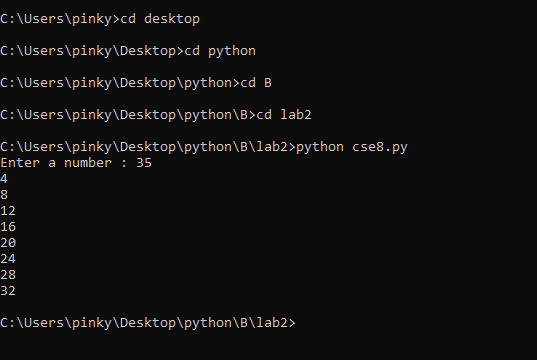
1. Enter a number : 30

4,8,12,14,16,20,24,28

2. Enter a number:45

4,8,12,14,16,20,24,28,32,36,40,44

**Output:**



**Result:**

The python script to print all leap years from 1 to n is successfully done.

***Case Study 09:***

***Problem Statement:***

A Python script for displaying a sequence of Fibonacci series till a given number.

i. Finding and displaying a Fibonacci series in between given range.

ii. To display nth term of Fibonacci series of n terms.

***Problem Solution:***

1. Take a number as input.

2. Initialize i and j to 1.

3. Perform the sum of i and j and print the result and change i and j values

4. Exit.

***Program/Source Code:***

#Python program to print fibonacci series

"""

Case study : 09

File name : cse9.py

Topics : Conditional and Looping statements

"""

#finding the fibbonaci numbers from 1 to given range

#taking the range value from the user

r=int(input(“enter the range value”))

print(“The fibbonaci numbers between 1 and {0} are”.format(r))

print

a=0

b=1

if a<r:

print(a)

if b<r:

print(b)

for i in range(r):

f=a+b

if f<r:

print(f)

else:

break

a=b

b=f

#finding fibbonaci numbers between m and n

#taking the both range values from the user

m=int(input(“enter the min value”))

n=int(input(“enter the max value”))

Print(“The fibbonaci numbers between {0} and {1} are”.format(m,n))

c=0

d=1

if c<n and c>m:

print(c)

if d<n and d>m:

print(d)

while True:

e=c+d

if e<n and e>m:

print(e)

c=d

d=e

elif e>=n:

break

else:

c=d

d=e

#printing the nth fibbonaci number in the series of n numbers

p=int(input(“Enter the n value:”))

a=0

b=1

if p==1:

print(a)

elif p==2:

print(b)

else:

for i in range(2,p):

f=a+b

a=b

b=f

print(f)

***Program Explanation:***

1. Take the input number and store it variable n.

2. Initialize i and j to 1.

3. Display the value of sum of i and j and assign i to j and j to the sum.

4. Repeat the step 3 until the sum is greater than or equal to n.

5. Exit.

***Runtime Test Cases:***

enter the range value 20

The fibbonaci numbers between 1 and 20 are

0

1

1

2

3

5

8

13

enter the min value 2

enter the max value 10

The fibbonaci numbers between 2 and 10 are

3

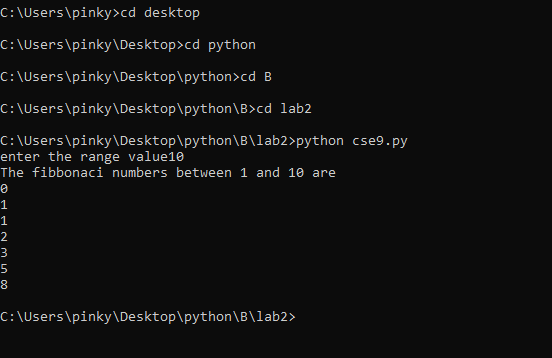
5

8

Enter the n value 8

The nth fibbonaci number is 13

**Output:**



**Result:**

The python script to display fibbonacci numbers upto a given value is successfully done.

***Case Study 10:***

***Problem Statement:***

Develop a text-based game that asks the user to guess a number! The system then outputs whether the guess was right or wrong! Let’s make the program give some help to the user! i.e. output whether the guess was higher or lower than the secret number!

***Problem Solution:***

1. Generate a random number.

2. Prompt the user to guess the value.

3. If it is greater than ask for another guess by displaying “high”.

4. Else if the guess is lower display “low” and ask for another guess.

5. Else if the guess is equal to the number then display the result.

6. Exit.

***Program/Source Code:***

#Python script for guessing a randomly generated number

"""

Case study : 10

File name : cse10.py

Topics : Conditional suite importing random, using randrange

"""

import random

#import random

#creation of variales

guess=0

tries=0

win=False

num=0

#take input from user

num=int(input("Enter a number to guess b/w 1 to 100:"))

#system random number

guess=random.randrange(1,100)

#demonstration of while suite

while(win):

if num>guess:

tries += 1

print("Too High")

elif num<guess:

tries +=1

print("Too low")

elif num==guess:

print("you won")

win=False

print(“you guessed it!”)

***Program Explanation:***

1. Import the random module to help generate a random number.

2. Use the randrange method to generate a random number by specifying the range and store it in random\_num.

3. Prompt the user to enter the guess a number.

4. If the guess is higher than random\_num then display “high” and ask for another guess.

5. Else if the guess is lower than random\_num then display “low” and ask for another guess.

6. Else display that the guess is correct.

7. Exit.

***Runtime Test Cases:***

1. Enter a number to guess b/w 1 to 100: 20

Too High

Enter a number to guess b/w 1 to 100: 10

Too High

Enter a number to guess b/w 1 to 100: 4

Too High

Enter a number to guess b/w 1 to 100: 3

You Won

2. Enter a number to guess b/w 1 to 100: 50

Too High

Enter a number to guess b/w 1 to 100: 25

Too Low

Enter a number to guess b/w 1 to 100: 33

Too High

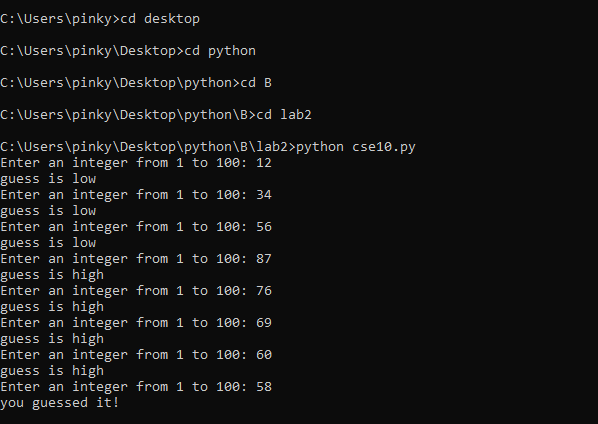
Enter a number to guess b/w 1 to 100: 30

Too High

Enter a number to guess b/w 1 to 100: 29

You Won

**Output:**



**Result:**

The python script to guess a random number is successfully done.

***Case Study 11:***

***Problem Statement:***

Write a python program to check whether given number is binary or not. A binary number is a number which contains only or 1. For example: 101101, 110010110, 10010011 are binary numbers.

***Problem Solution:***

1. Take an input from the user.

2. If the number contains any other digit other than 0 or 1 then it is not a binary number.

3. Check if it has any such that digits.

4. If yes display that it is not binary.

5. Else display it is a binary number.

6. Exit.

***Program/Source Code:***

#Python script to check whether a given number is binary or not

"""

Case study : 11

File name : cse11.py

Topics : Conditional Statements and while suite

"""

#input from the user

a = int(input("Enter a number :"))

b = a

flag = 0

while(b>0):

d = b % 10

if(d > 1):

flag=1

break

b = b // 10

if(flag == 1):

print("Given number is not binary")

else:

print("Given number is binary")

***Program Explanation:***

1. Take input from the user and store it in variable num.

2. Assign num value to num1.

3. Start checking if the digits of the number by using the modulo operator.

4. If all the digits are either 0s or 1s then display that the given number is binary number.

5. If it has digits other than 0s and 1s then display that it is not a binary number.

6. Exit.

***Runtime Test Cases:***

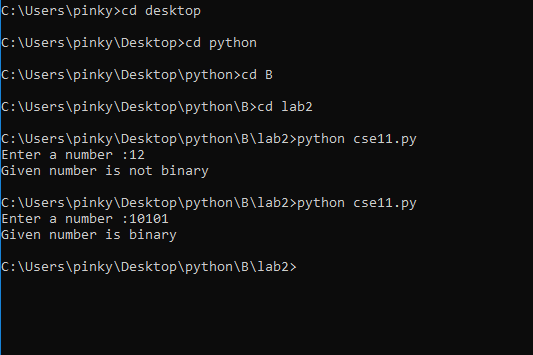
1. Enter a number:102002

Given number is not binary.

2. Enter a number:111111

Given number is binary.

**Output:**



**Result:**

The python script to check whether a given number is binary or not is successfully done.

***Case Study 12:***

***Problem Statement:***

Python program for calculation of the Pythagorean Numbers ( Pythagorean Triplets)

***Problem Solution:***

1. First we import math so we can use the square root with math.sqrt.

2. Create a function called pythagorean\_triplet that receives one argument (n)

3. Create a for loop ‘b’ in range(n)

4. Then we create a for nested loop ‘a’ that will loop from 1 to a value less than b.

5. Then we check of reminders of a division of one and we convert c to an integer because c is a float number.

***Program/Source Code:***

#Practice program for calculation of the Pythagorean Numbers ( Pythagorean Triplets)

"""

Case study : 12

File name : cse12.py

Topics : Conditional Statements for suite

"""

#importing math module

import math

n=int(input("Enter maximal Number"))

for i in range(1,n):

for j in range(1,i):

cs= i\*\*2 + j\*\*2

c=int(math.sqrt(cs))

if((c\*\*2-cs)==0):

print(i,j,c)

***Program Explanation:***

1. Take a number as input and pass it as argument to function pythagorean\_triplet(n).

2. A for loop for b in range(n) is created which will iterates until the b is in range(n).

3. A nested for loop for a in range(1,b) is created which iterates until a is in range(1,b).

4. Now calculate c2=a2+b2 (Pythagoras theorem) by using math.sqrt function.

5. Now if c%1 is equal to 0 then print a b and c values.

6. Exit.

***Runtime Test Cases:***

1.Enter max range: 6

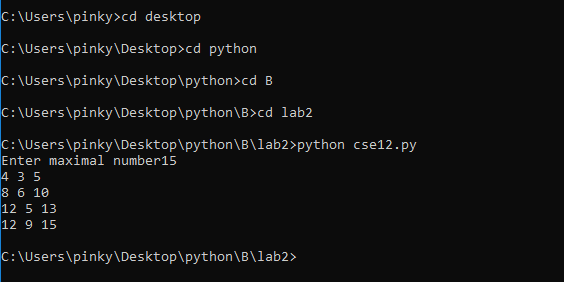
3 4 5

2.Enter max range: 11

3 4 5

6 8 10

**Output:**



**Result:**

Implementation of basic concepts of variables, operators, value types, statements, conditional statements and loops done successfully.

***Case Study 13:***

***Problem Statement:***

Write a program that prints the numbers from 1 to 20. But for multiples of three print “Fizz” instead of

the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three

and five print “FizzBuzz”.

***Problem Solution:***

Take input of a number.

Use range function to iterate from 1 to number using iterator i.

2.1. If i is divisible by 3 and 5 print FizzBuzz.

2.2. Else If i is divisible by 3 print Fizz.

2.3. Else If i is divisible by 5 print Buzz.

2.4. else print i.

***Program/Source Code:***

#Python program to do fizzbuzz game

"""

Case study : 13

File name : cse13.py

Topics : Conditional Statements and for suite

"""

#Taking input from user

num=int(input(“Enter a number”))

#Logic

for i in range(1,num):

if(i%3==0 and i%5==0):

print(“FizzBuzz”)

elif(i%3==0):

print(“Fizz”)

elif(i%5==0):

print(“Buzz”)

else:

print(i)

***Program Explanation:***

Take input of a number and store it in num.

Use range function to iterate from 1 to num using iterator i.

2.1. If i is divisible by 3 and 5 print FizzBuzz.

2.2. Else If i is divisible by 3 print Fizz.

2.3. Else If i is divisible by 5 print Buzz.

2.4. else print i.

3.Exit

***Runtime Test Cases:***

1. Enter a number : 20

1

2

Fizz

4

Buzz

Fizz

7

8

Fizz

Buzz

11

Fizz

13

14

FizzBuzz

16

17

Fizz

19

1. Enter a number : 5

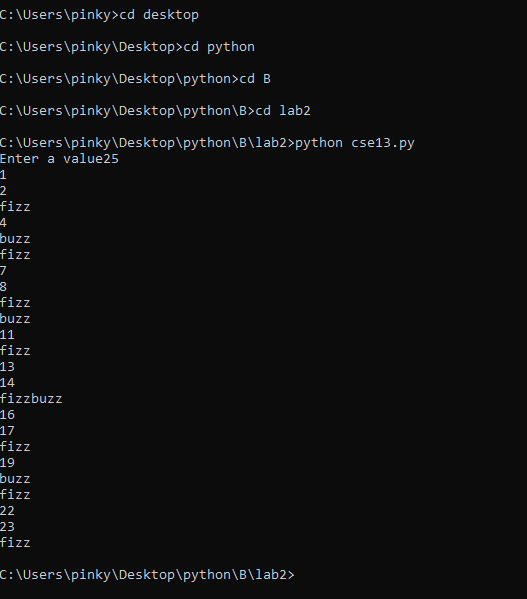
1

2

Fizz

4

**Output:**



**Result:**

Implementation of basic concepts of variables, operators, value types, statements, conditional statements

and loops done successfully.

***Case Study 14:***

***Problem Statement:***

A Python Program to Calculate Electricity Bill.

***Problem Solution:***

1. Take input of no. of Watts consumed per hour daily for one month

2. Convert Watts consumed into Units.

3. Calculate the total energy bill of that consumer in Rs.

If Units consumed is less than 100, fix the rate per unit as 3? (In Rs.)

If Units consumed is in between 100 and 200, fix the rate per unit as

5? (In Rs.)

If Units consumed is greater than 200, fix the rate per unit as 7? (In

Rs.)

[Take 1 month = 30 Days].

***Program/Source Code:***

#Practice program to calculate the electricity bill using if else statements

"""

Case study : 14

File name : cse14.py

Topics : Variable, Values and types and

"""

#Take input of no. of Watts consumed per hour daily for one month

Watt=int(input(“Enter the no of watts:”))

#Converting watts into units

units=Watt\*24\*30

units=units//1000

#caluclating the total energy bill

if units<100:

rate=units\*3

elif units>=100 and units<=200:

rate=units\*5

elif units>200:

rate=units\*7

print(“The total electricity bill is”,rate,”Rs”)

***Program Explanation:***

1. Take a number as input and store it in the variable noOfWatts.

2. Choose a variable p and store the product of noOfWatts , noOfHours(24), NoOfDays(30).

3. Units Consumed is calculated by assigning variable consumed with the  value of p changed into units

from watts

4. Using the if else condition check if the consumed units  is &lt; than 100 or not .If it is &lt; 100 then assign

rate = consumed \*3 and print rate value.

5.  elif condition check if the consumed units  is between 100 and 200 . If it is between 100 and 200

 then assign rate = consumed \*5 and print rate value

6.else assign rate = consumed.\*7 and print rate value.

7. Otherwise print the output as “It is not a prime number” and exit.

***Runtime Test Cases:***

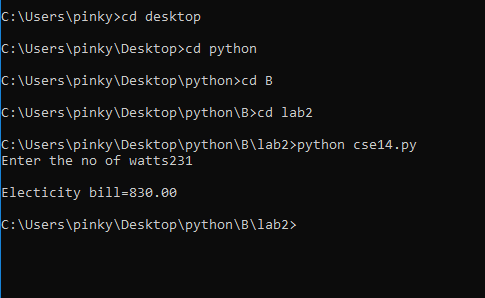
1. Enter the no of watts 890

The total electricity bill is 4480 Rs

1. Enter the no of watts 1000

The total electricity bill is 5040 Rs

**Output:**



**Result:**

Implementation of basic concepts of variables, operators, value types, statements, conditional statements

and loops done successfully.