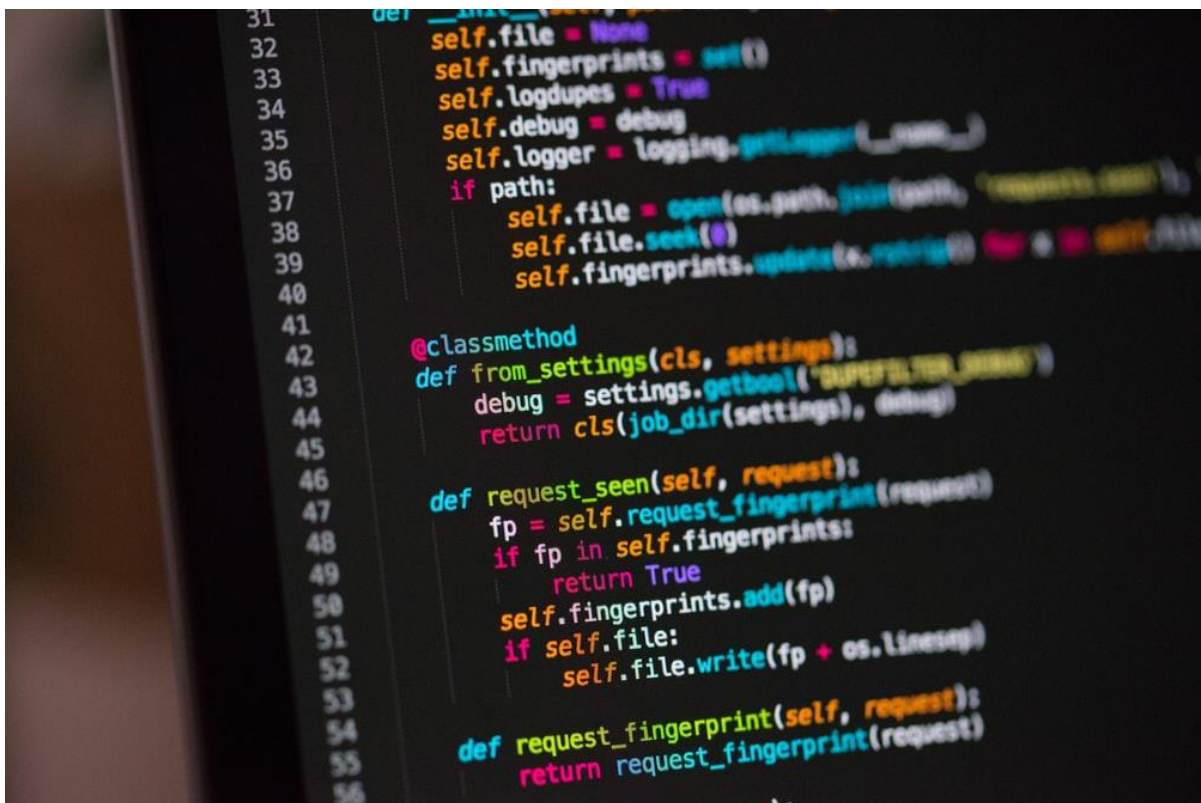




SYMBIOSIS INSTITUTE OF TECHNOLOGY (SIT)

Programming and Problem Solving – Lab



Experiments Record - Journal

~ Mudit Garg

CSE B2

Batch 2021-25

PRN: 21070122098

Experiment No. 1-A

Title:

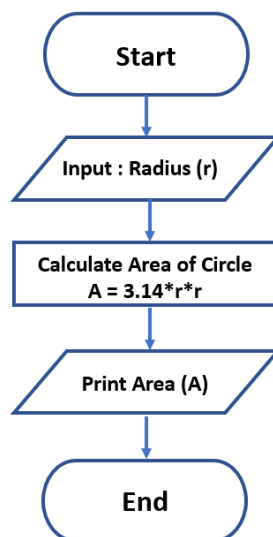
To design and develop a flowchart and an algorithm for finding the area of a circle, of a rectangle, of a square and of a triangle.

Tool/Platform:

Microsoft Word / PowerPoint

Flowcharts and Algorithms:

Flowchart – Area of a Circle

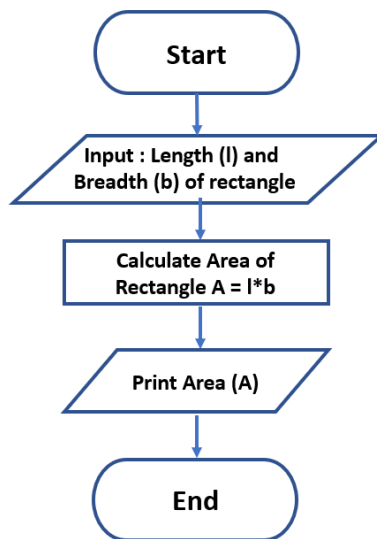


13/10/2021

Algorithm for Area of Circle:

1. Initialise the program
2. Initialise variable “r” and assign it the value as entered by user for radius of the circle
3. Initialise variable “a”
4. Calculate the area of circle using the formula, **area = 3.14*r*r** and assign it to the variable “a”
5. Print the area of circle, “a” as the output
6. End the program

Flowchart – Area of a Rectangle

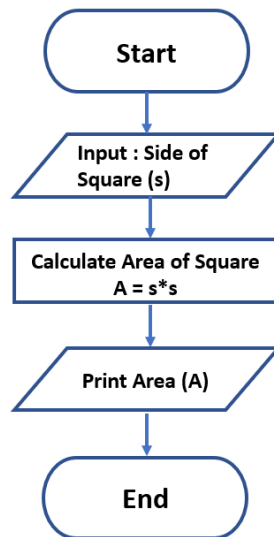


13/10/2021

Algorithm for Area of Rectangle:

1. Initialise the program
2. Initialise variable “l” and “b” and assign them a value as entered by user for length and breadth of the rectangle
3. Initialise variable “a”
4. Calculate the area of rectangle using the formula, **area = l*b** and assign it to the variable “a”
5. Print the area of rectangle, “a” as the output
6. End the program

Flowchart – Area of a Square

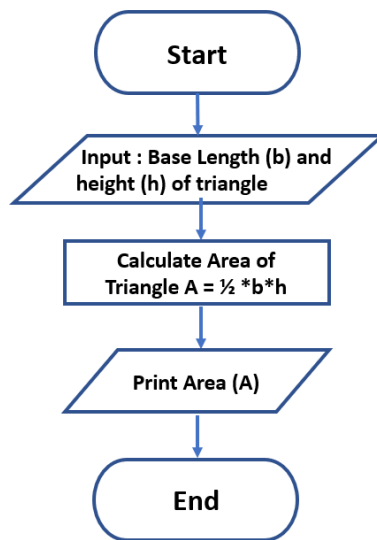


13/10/2021

Algorithm for Area of Square:

1. Initialise the program
2. Initialise variable “s” and assign it the value as entered by user for side of the square
3. Initialise variable “a”
4. Calculate the area of square using the formula, **area = s*s** and assign it to the variable “a”
5. Print the area of square, “a” as the output
6. End the program

Flowchart – Area of a Triangle



13/10/2021

Algorithm for Area of Triangle:

1. Initialise the program
2. Initialise variable “b” and “h” and assign it the value as entered by user for the base length and height of the triangle
3. Initialise variable “a”
4. Calculate the area of triangle using the formula, **area = $\frac{1}{2} * b * h$** and assign it to the variable “a”
5. Print the area of triangle, “a” as the output
6. End the program

Learning Outcome:

By performing this experiment, I was able to understand the concept of Computational Thinking and implement it. I was able to understand that why algorithms and flowcharts are important in solving a problem and how they help in achieving a better result faster.

Experiment No. 1-B

Title:

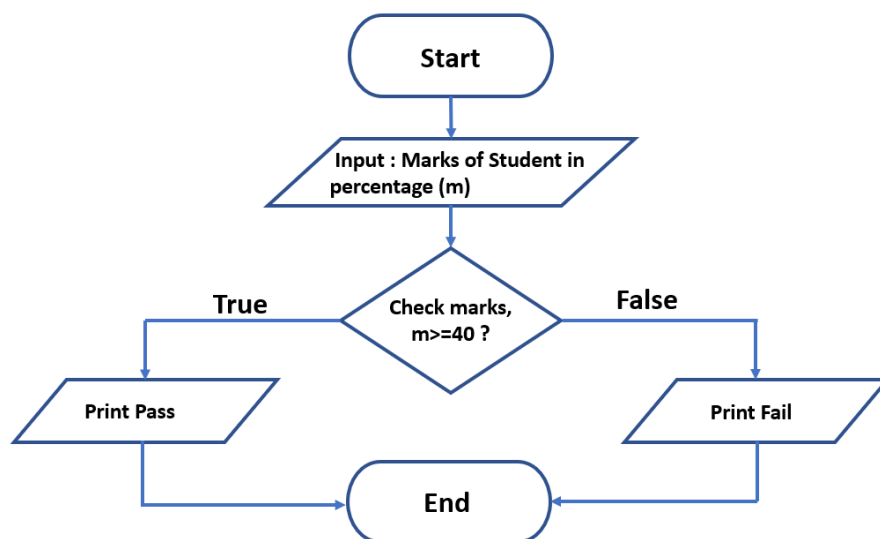
To design and develop a flowchart and an algorithm to determine whether a student has passed the exam or not

Tool/Platform:

Microsoft Word / PowerPoint

Flowchart:

Flowchart – Student Examination



13/10/2021

Algorithm:

1. Initialise the program
2. Initialise variable “m” and assign it the value as entered by user for marks obtained out of 100
3. Check if marks are greater than 40 or not
4. If marks are greater than 40, Print “**Pass**” as the output
5. Else print “**Fail**” as the output
6. End the Program

Learning Outcome:

By performing this experiment, I was able to understand the concept of Computational Thinking and implement it. I was able to understand that why

algorithms and flowcharts are important in solving a problem and how they help in achieving a better result faster.

Experiment No. 2

Title:

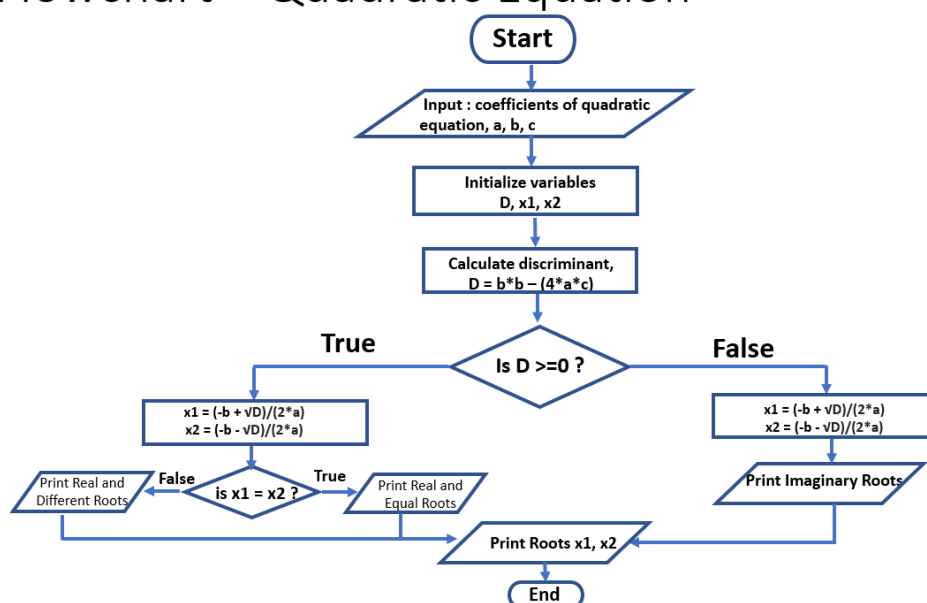
To design and develop a flowchart and an algorithm that takes three coefficients a , b and c of a quadratic equation $ax^2 + bx + c = 0$ as input and compute all possible roots of the equation

Tool/Platform:

Microsoft Word / PowerPoint

Flowchart:

Flowchart – Quadratic Equation



20/10/2021

Algorithm:

1. Initialize Program
2. Initialize variables “ a ”, “ b ”, “ c ” and assign the value as per input from user for coefficients of quadratic equation
3. Initialize variables “ D ”, “ $x1$ ”, “ $x2$ ”
4. Calculate discriminant as $b*b - (4*a*c)$ and assign to “ D ”
5. If $D \geq 0$, calculate roots as “ $x1 = (-b + \sqrt{D})/(2*a)$ ”, “ $x2 = (-b - \sqrt{D})/(2*a)$ ”
6. If $x1 = x2$, print “Real and Equal Roots”
7. Else, if $D < 0$, calculate roots as $x1 = (-b + \sqrt{D})/(2*a)$, $x2 = (-b - \sqrt{D})/(2*a)$, Print “Imaginary/ Complex Roots”
8. Print Roots “ $x1$ ”, “ $x2$ ”

Learning Outcome:

By performing this experiment, I was able to understand the concept of Computational Thinking and implement it. I was able to understand that why algorithms and flowcharts are important in solving a problem and how they help in achieving a better result faster and how to break a complex problem into simpler parts and solve to get the answer easily and efficiently.

Experiment No. 3-A

Title:

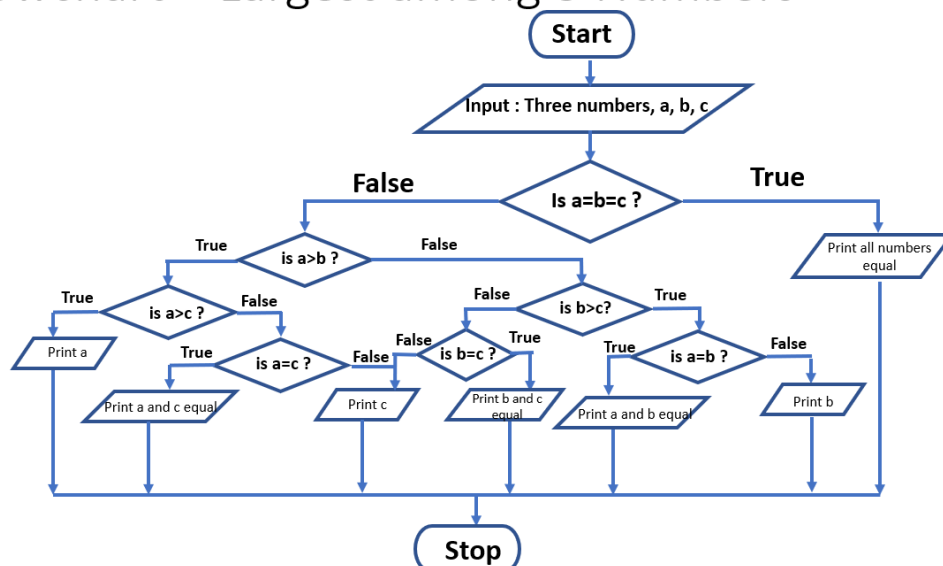
To design and develop a flowchart and an algorithm that takes three numbers a, b and c as input and prints the value of the largest number

Tool/Platform:

Microsoft Word / PowerPoint

Flowchart:

Flowchart – Largest among 3 Numbers



27/10/2021

Algorithm:

1. Initialize Program
2. Initialize variables “a”, “b”, “c” and assign the value as per input from user for three numbers
3. Check if the all the numbers are **equal or not**
4. If all the numbers are equal, print “**All numbers equal**”, and their **value**
5. Else, check if **a>b?**
6. If a>b check for **a>c**
7. If a>c, print “a” as the output, else check for **a=c**
8. If a=c, print “**a and c equal**”, and their **value**, else print “c”
9. If a>b is false, check for **b>c**
10. If b>c is false, check for **b=c** and print “**b and c equal**” and their **value if true** or else print “c”
11. If b>c is true check for **b=a**
12. If b=a, print “**a and b equal**”, and their value, else print “b”

13. Stop the program

Learning Outcome:

By performing this experiment, I was able to understand the concept of Computational Thinking and implement it. I was able to understand that why algorithms and flowcharts are important in solving a problem and how they help in achieving a better result faster and how to break a complex problem into simpler parts and solve to get the answer easily and efficiently.

Experiment No. 3-B

Title:

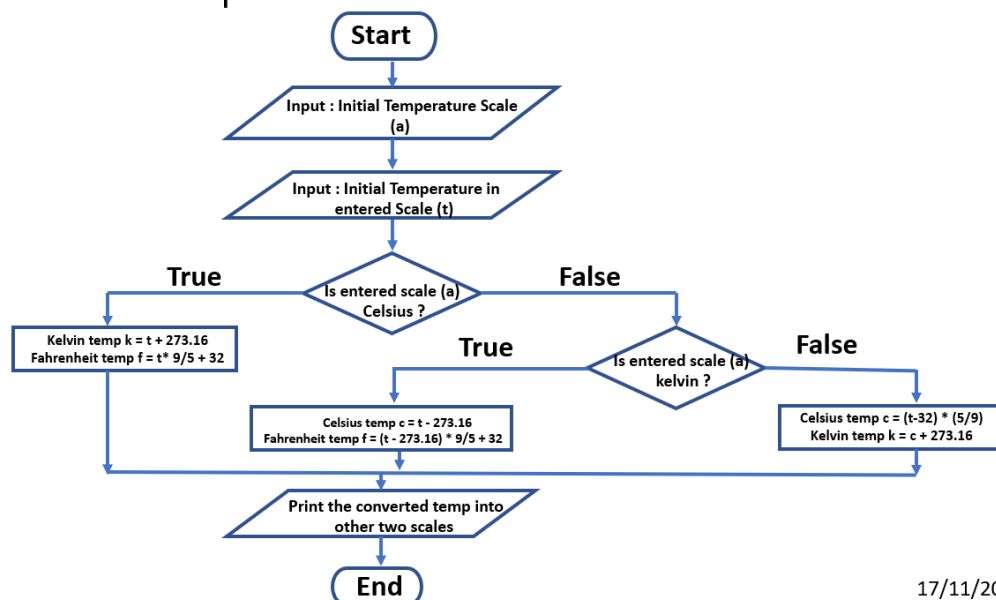
Write a Python Program to convert the entered temperature in Celsius into other two temperature scales.

Tool/Platform:

Microsoft Word / PowerPoint and Python IDLE

Flowchart:

Flowchart – Temperature Conversion



17/11/2021

Algorithm:

1. Initialise the program
2. Initialise variable "a" and "t"
3. Input Initial Temperature Scale from User and assign it to "a"
4. Input Temperature and assign it to "t"
5. Check if entered scale "a" is Celsius
6. If yes, calculate temperature in Fahrenheit using $f = t * (9/5) + 32$ and in Kelvin using $t + 273.16$
7. Else if false, check for Scale being Kelvin

8. If true, calculate temperature in Fahrenheit using $f = (t - 273.16) * (9/5) + 32$ and in Celsius using $t - 273.16$
9. If the entered scale is not Kelvin, it is Fahrenheit
10. Calculate the temperature in Celsius as $c = (t - 32) * (5/9)$ in Kelvin as $k = c + 273.16$
11. Print the converted temperature with scales
12. End the program

Source Code:

```
# Temperature Conversion
# Date 17/11/21
import sys

print("Temperature Conversion from One to another scale")

a = input("Select Default Temperature Scale from Celsius / Fahrenheit / Kelvin: ")
if a not in ["Celsius", "celsius", "Fahrenheit", "fahrenheit", "Kelvin", "kelvin"]:
    sys.exit("Invalid Input, Recheck !")

print("You have selected", a, "as your default temperature measurement scale!")
t = eval(input(" Enter the temperature in Selected Scale : "))

if a == "Celsius" or a == "celsius":
    k = t + 273.16
    f = t*(9/5) + 32
    print(f' The entered temperature {t} converted from {a} to Kelvin is', round(k,2), "and in Fahrenheit is", round(f,2), ' !')

elif a == "Fahrenheit" or a == "fahrenheit":
    c = (t - 32) * (5/9)
    k = c + 273.16
    print(f' The entered temperature {t} converted from {a} to Kelvin is', round(k,2), "and in Celsius is", round(c,2), ' !')

elif a == "Kelvin" or a == "kelvin":
    c = t - 273.16
    f = (t - 273.16) * (9/5) + 32
    print(f' The entered temperature {t} converted from {a} to Celsius is', round(c,2), "and in Fahrenheit is", round(f,2), ' !')

else:
    print("Invalid Operations, Recheck !")
```

```

# Temperature Conversion
# Date 17/11/21
import sys

print("Temperature Conversion from One to another scale")

a = input("Select Default Temperature Scale from Celsius / Fahrenheit / Kelvin: ")
if a not in ["Celsius", "celsius", "Fahrenheit", "fahrenheit", "Kelvin", "kelvin"]:
    sys.exit("Invalid Input, Recheck !")

print("You have selected", a, "as your default temperature measurement scale!")
t = eval(input(" Enter the temperature in Selected Scale : "))

if a == "Celsius" or a == "celsius":
    k = t + 273.16
    f = t*(9/5) + 32
    print(f' The entered temperature {t} converted from {a} to Kelvin is', round(k,2)," and in Fahrenheit is", round(f,2),' !')

elif a == "Fahrenheit" or a == "fahrenheit":
    c = (t - 32) * (5/9)
    k = c + 273.16
    print(f' The entered temperature {t} converted from {a} to Kelvin is', round(k,2)," and in Celsius is", round(c,2),' !')

elif a == "Kelvin" or a == "kelvin":
    c = t - 273.16
    f = (t - 273.16) * (9/5) + 32
    print(f' The entered temperature {t} converted from {a} to Celsius is', round(c,2)," and in Fahrenheit is", round(f,2),' !')

else:
    print("Invalid Operations, Recheck !")

```

Output Screenshot:

```

Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\3. Temperature Conversion.py
Temperature Conversion from One to another scale
Select Default Temperature Scale from Celsius / Fahrenheit / Kelvin: Kelvin
You have selected Kelvin as your default temperature measurement scale!
Enter the temperature in Selected Scale : 300
The entered temperature 300 converted from Kelvin to Celsius is 26.84 and in Fahrenheit is 80.31 !
>>> |

```

```

Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\3. Temperature Conversion.py
Temperature Conversion from One to another scale
Select Default Temperature Scale from Celsius / Fahrenheit / Kelvin: celsius
You have selected celsius as your default temperature measurement scale!
Enter the temperature in Selected Scale : -40
The entered temperature -40 converted from celsius to Kelvin is 233.16 and in Fahrenheit is -40.0 !
>>> |

```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\3. Temperature Conversion.py
Temperature Conversion from One to another scale
Select Default Temperature Scale from Celsius / Fahrenheit / Kelvin: fahrenheit
You have selected fahrenheit as your default temperature measurement scale!
Enter the temperature in Selected Scale : 104
The entered temperature 104 converted from fahrenheit to Kelvin is 313.16 and in Celsius is 40.0 !
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\3. Temperature Conversion.py
Temperature Conversion from One to another scale
Select Default Temperature Scale from Celsius / Fahrenheit / Kelvin: cel
SystemExit: Invalid Input, Recheck !
>>>
```

Learning Outcome:

By performing this experiment, I was able to understand the basics concepts and syntax of Python Programming Language and write a code in it. The use of built-in functions and loop was clear. The basic data types in Python were understood and used to get appropriate result. Functions, Control flow statements and different data types were used to represent data, process it and give the required output. Use of if-else or if-elif-else loop or the nested if-else loop was understood and the syntax was clear. Its function was understood and it was used to get the desired result.

Experiment No. 4

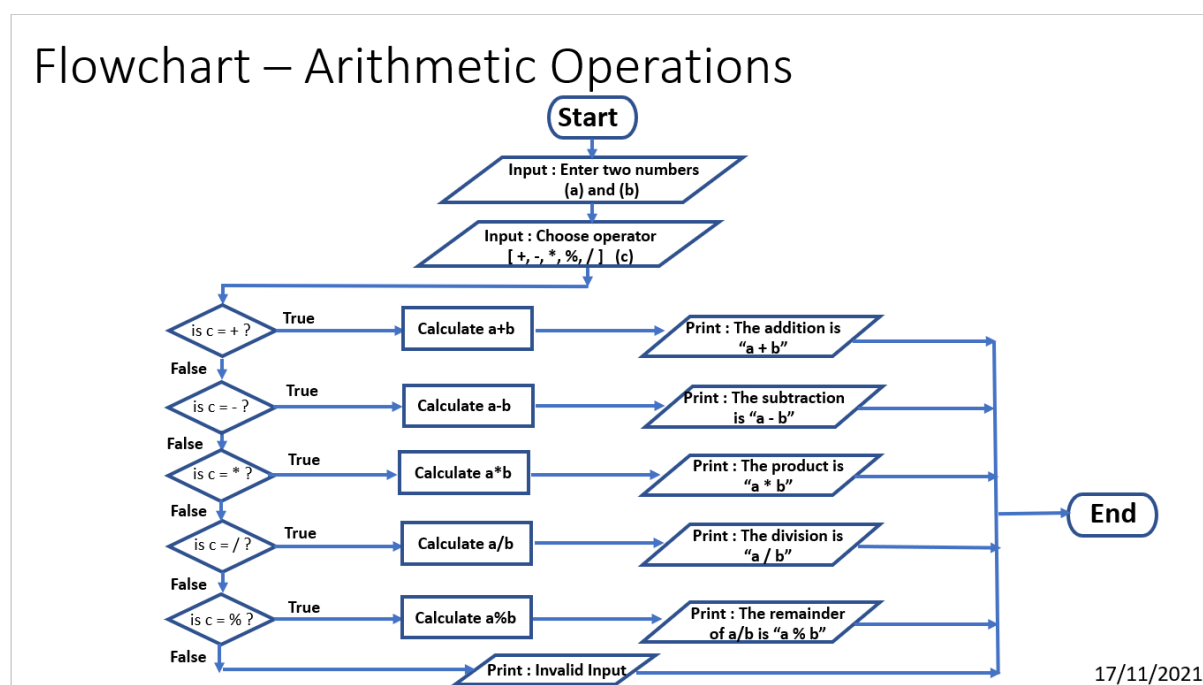
Title:

Write a Python Program to perform different arithmetic operations like in mathematics.

Tool/Platform:

Microsoft Word / PowerPoint and Python IDLE

Flowchart:



Algorithm:

1. Initialise the program
2. Initialise variable "**a**", "**b**" and "**c**"
3. Input two numbers from user and assign them to "**a**" and "**b**"
4. Input the Operation to be performed and assign it to "**c**"
5. Check if "**c**" is "+" and if true perform the operation "**a+b**" and print the output
6. If false, check if "**c**" is "-" and if this is true perform the operation "**a-b**" and print the output
7. Else, check if "**c**" is "*" and if this is true perform the operation "**a*b**" and print the output
8. If false, check if "**c**" is "/" and if this is true perform the operation "**a/b**" and print the output
9. Else, check if "**c**" is "%" and if this is true perform the operation "**a%b**" and print the output
10. Else, if false, Print "**Invalid Operation**"

11. End the program

Source Code:

```
#if - else condition to perform simple arithmetic functions for entered two values
#Date 17/11/2021
a=eval(input("Enter a number: "))
b=eval(input("Enter second number with which to perform the simple arithmetic
operation: "))
c=input("Choose an operator [ + , - , / , * , % [To get remainder when a is divided by
b] ")
if c == '+' :
    print("The addition of two entered numbers is : " , a+b)
elif c == '-' :
    print("The subtraction of two entered numbers is : " , a-b)
elif c == '*' :
    print("The multiplication of two entered numbers is : " , a*b)
elif c == '/' :
    print("The division of two entered numbers is : " , a / b)
elif c == '%' :
    print("The remainder when a is divided by b is:" , a%b)
else:
    print("Entered Operator is INVALID !!!")
```

```
#if - else condition to perform simple arithmetic functions for entered two values
#Date 17/11/2021
a=eval(input("Enter a number: "))
b=eval(input("Enter second number with which to perform the simple arithmetic operation: "))
c=input("Choose an operator [ + , - , / , * , % [To get remainder when a is divided by b ] ")
if c == '+' :
    print("The addition of two entered numbers is : " , a+b)
elif c == '-' :
    print("The subtraction of two entered numbers is : " , a-b)
elif c == '*' :
    print("The multiplication of two entered numbers is : " , a*b)
elif c == '/' :
    print("The division of two entered numbers is : " , a / b)
elif c == '%' :
    print("The remainder when a is divided by b is:" , a%b)
else:
    print("Entered Operator is INVALID !!!")
```

Output Screenshots:

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\1. WAP to perform all mathematical functions.py
Enter a number: 12
Enter second number with which to perform the simple arithmetic operation: 2
Choose an operator [ + , - , / , * , % [To get remainder when a is divided by b ] +
The addition of two entered numbers is : 14
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\1. WAP to perform all mathematical functions.py
Enter a number: 14
Enter second number with which to perform the simple arithmetic operation: 37
Choose an operator [ + , - , / , * , % [To get remainder when a is divided by b ] -
The subtraction of two entered numbers is : -23
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\1. WAP to perform all mathematical functions.py
Enter a number: 5
Enter second number with which to perform the simple arithmetic operation: 13
Choose an operator [ + , - , / , * , % [To get remainder when a is divided by b ] *
The multiplication of two entered numbers is : 65
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\1. WAP to perform all mathematical functions.py
Enter a number: 60
Enter second number with which to perform the simple arithmetic operation: 5
Choose an operator [ + , - , / , * , % [To get remainder when a is divided by b ] /
The division of two entered numbers is : 12.0
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\1. WAP to perform all mathematical functions.py
Enter a number: 100
Enter second number with which to perform the simple arithmetic operation: 3
Choose an operator [ + , - , / , * , % [To get remainder when a is divided by b ] %
The remainder when a is divided by b is: 1
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\1. WAP to perform all mathematical functions.py
Enter a number: 12
Enter second number with which to perform the simple arithmetic operation: 34
Choose an operator [ + , - , / , * , % [To get remainder when a is divided by b ] .
Entered Operator is INVALID !!!
>>>
```

Learning Outcome:

By performing this experiment, I was able to understand the basics concepts and syntax of Python Programming Language and write a code in it. The use of built-in functions and loop was understood. The basic data types in Python were used to get appropriate result. Use of in-built arithmetic operators was understood. Inbuilt arithmetic functions and if-else loop were used to get required result.

Experiment No. 5-A

Title:

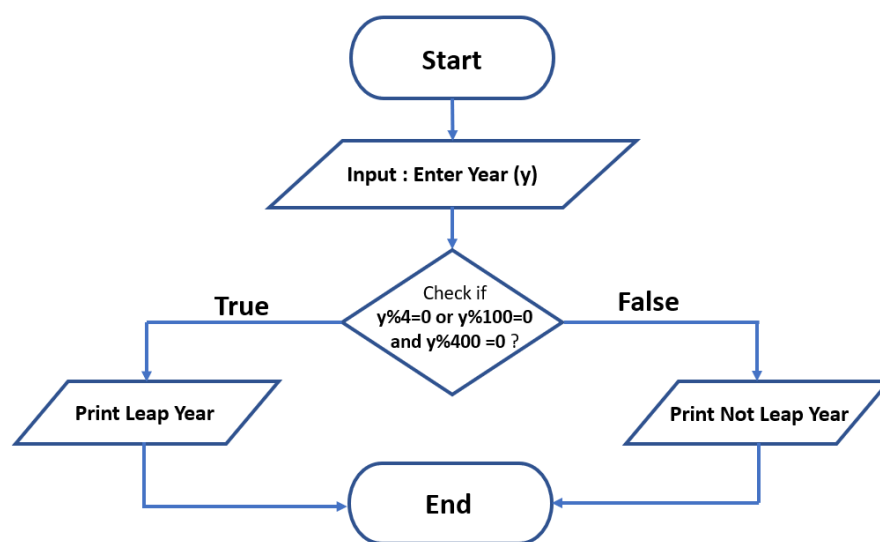
Write a Python Program to check if the given year is a leap year.

Tool/Platform:

Microsoft Word / PowerPoint and Python IDLE

Flowchart:

Flowchart – Check if Leap Year



17/11/2021

Algorithm:

1. Initialise the program
2. Initialise variable "y"
3. Input value of Year from user and assign it to "y"
4. Check if **y%4=0 or y%100=0 and y%400=0**
5. If True, Print "Leap Year"
6. Else, Print "Not a Leap Year"

Source Code:

```
#Check if leap year or not  
# Date 17/11/21
```

```
year = int(input("Please Enter the Year Number you wish to check: "))
```

```
if (year%4 == 0) or (year%100 == 0) and (year%400 == 0):
```

```
print(f' Entered year {year} is a Leap Year')
else:
    print(f' Entered year {year} is not a Leap Year')
```

```
#Check if leap year or not
# Date 17/11/21

year = int(input("Please Enter the Year Number you wish to check: "))

if (year%4 == 0) or (year%100 == 0) and (year%400 == 0):
    print(f' Entered year {year} is a Leap Year')
else:
    print(f' Entered year {year} is not a Leap Year')
```

Output Screenshots:

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\4. Check if leap year or not.py
Please Enter the Year Number you wish to check: 2020
Entered year 2020 is a Leap Year
>>>|
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\4. Check if leap year or not.py
Please Enter the Year Number you wish to check: 2021
Entered year 2021 is not a Leap Year
>>>|
```

Learning Outcome:

By performing this experiment, I was able to understand the basics concepts and syntax of Python Programming Language and write a code in it. The basic data types in Python were used to get appropriate result. Use of if-else loop was understood. Use of Arithmetic Operators and Logical Operators was made clear. Syntax of “f- string formatting” was understood and it was used to get required result.

Experiment No. 5-B

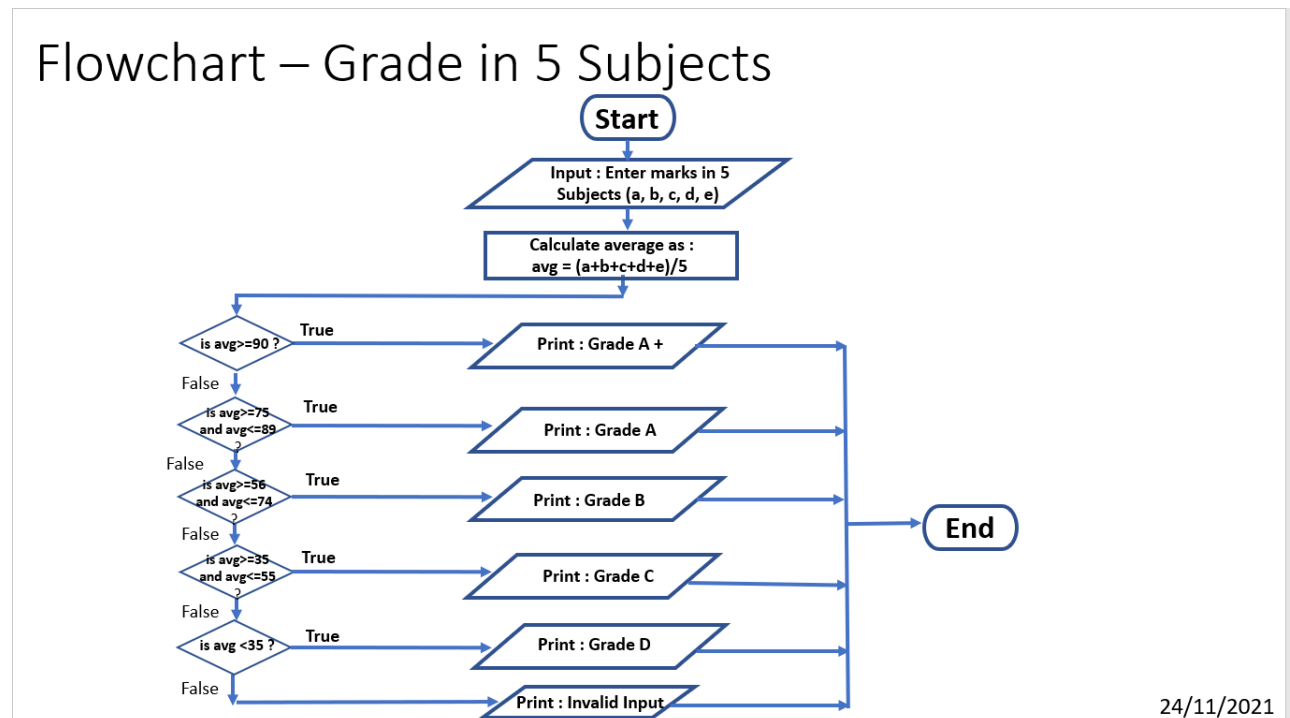
Title:

Write a Python Program to input Marks in 5 Subjects and Display the Grade.

Tool/Platform:

Microsoft Word / PowerPoint and Python IDLE

Flowchart:



Algorithm:

1. Initialise the program
2. Initialise variable “a”, “b”, “c”, “d”, “e” and “avg”
3. Input marks in 5 subjects from user and assign them to “a”, “b”, “c”, “d”, “e”
4. Calculate Average and assign it to “avg”
5. Check if “avg” is “>=90” and if true print “Grade A+”
6. If false, check if “avg” is “>=75 and <=89” and if this is true print “Grade A”
7. Else, check if “avg” is “>=56 and <=74” and if this is true print “Grade B”
8. If false, check if “avg” is “>=35 and <=55” and if this is true print “Grade C”
9. Else, check if “avg” is “<35” and if this is true print “Grade D”
10. Else, if false, Print “Invalid Operation”
11. Exit

Source Code:

```
# Grade based on marks in 5 subjects
# Date : 24/11/2021
import sys

s1= float(input("Enter Marks Scored in Mathematics: "))
s2= float(input("Enter Marks Scored in Science: "))
s3= float(input("Enter Marks Scored in English: "))
s4= float(input("Enter Marks Scored in Hindi: "))
s5= float(input("Enter Marks Scored in Social Science: "))
print()
avg= (s1 + s2 + s3 + s4 + s5)/5
print("Scored Percentage:", avg)
if avg >= 90 :
    print(" Outstanding : Grade A* ")
elif avg>= 75 and avg <=89:
    print(" Excellent : Grade A ")
elif avg>= 56 and avg <=74:
    print(" Very Good : Grade B ")
elif avg>= 35 and avg <=55:
    print(" Good : Grade C ")
elif avg <35:
    print(" Scope for Improvement : Grade D ")
else:
    sys.exit("Invalid Input, Recheck !")
```

```
# Grade based on marks in 5 subjects
# Date : 24/11/2021
import sys

s1= float(input("Enter Marks Scored in Mathematics: "))
s2= float(input("Enter Marks Scored in Science: "))
s3= float(input("Enter Marks Scored in English: "))
s4= float(input("Enter Marks Scored in Hindi: "))
s5= float(input("Enter Marks Scored in Social Science: "))
print()
avg= (s1 + s2 + s3 + s4 + s5)/5
print("Scored Percentage:", avg)
if avg >= 90 :
    print(" Outstanding : Grade A* ")
elif avg>= 75 and avg <=89:
    print(" Excellent : Grade A ")
elif avg>= 56 and avg <=74:
    print(" Very Good : Grade B ")
elif avg>= 35 and avg <=55:
    print(" Good : Grade C ")
elif avg <35:
    print(" Scope for Improvement : Grade D ")
else:
    sys.exit("Invalid Input, Recheck !")
```

Output Screenshots:

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\6. Marks in 5 subjects.py
Enter Marks Scored in Mathematics: 100
Enter Marks Scored in Science: 95
Enter Marks Scored in English: 96
Enter Marks Scored in Hindi: 90
Enter Marks Scored in Social Science: 99

Scored Percentage: 96.0
Outstanding : Grade A*
```

Learning Outcome:

By performing this experiment, use of if-elif-else loop was understood. Use of Arithmetic Operators and Logical Operators was made clear. Appropriate data types were used to perform required operations and thus get required result.

Experiment No. 6-A

Title:

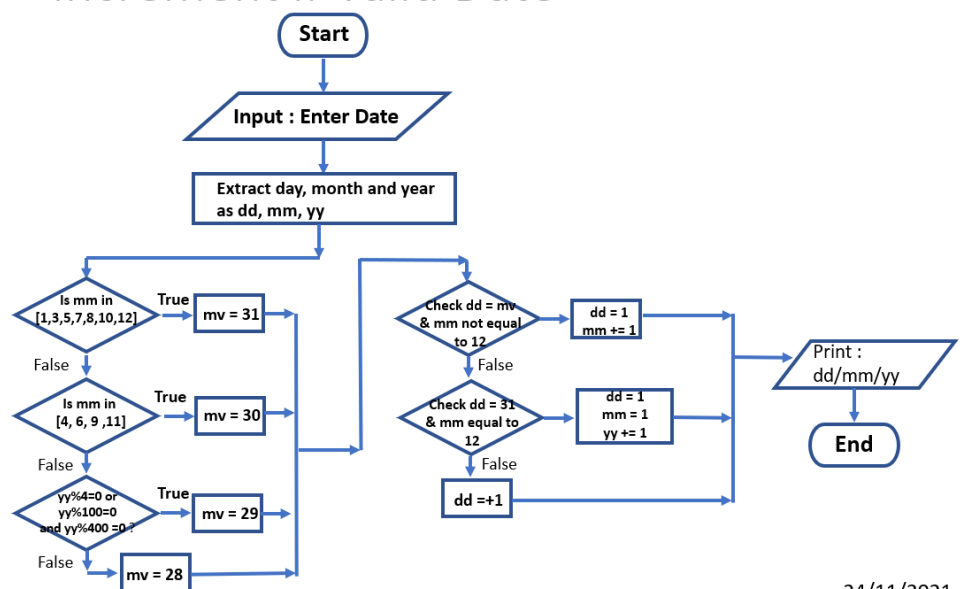
Write a Python Program to check if a date is valid and print the incremented date if it is.

Tool/Platform:

Microsoft Word / PowerPoint and Python IDLE

Flowchart:

Flowchart – Increment if Valid Date



24/11/2021

Algorithm:

1. Initialise the program
2. Initialise variable “**date**”, “**dd**”, “**mm**”, “**yy**”, “**mv**”
3. Input date from user and assign to “**date**”
4. Extract day, month and year and assign to “**dd**”, “**mm**”, “**yy**”
5. Check for maximum number of days in given month and assign to “**mv**”
6. If “**dd**” is equal to “**mv**” and “**mm**” is not equal to 12, set “**dd**” as 1 and increment value of “**mm**” by 1
7. Else, check if “**dd**” is equal to 31 and “**mm**” is equal to 12, if true, set “**dd**” = 1, “**mm**” = 1 and increment value of “**yy**” by 1
8. If false, increment value of “**dd**” by 1
9. Print the incremented date
10. Exit

Source Code:

```
#Check if entered date is in valid form
#Print Incremented Date
#Date : 24/11/2021

from datetime import *

idate=input("Enter date in the format dd/mm/yyyy: ")
dd,mm,yyyy=idate.split('/')
dd=int(dd)
mm=int(mm)
yyyy=int(yyyy)
if mm in [1,3,5,7,8,10,12]:
    mv= 31
elif mm in [4,6,9,11]:
    mv= 30
elif (yyyy%4==0 and yyyy%100!=0 or yyyy%400==0):
    mv= 29
else:
    mv= 28

if mm < 1 or mm > 12 or dd < 1 or dd > mv :
    sys.exit("Invalid Date Entered, Recheck !")
elif dd==mv and mm!=12:
    dd= 1
    mm+= 1
elif dd== 31 and mm == 12:
    dd = 1
    mm = 1
    yyyy +=1
else:
    dd +=1

odate= date(yyyy,mm,dd)
fdate=odate.strftime("%d/%m/%Y")
print("Entered Date:" , idate)
print("Incremented Date: ",fdate)
```

```

#Check if entered date is in valid form
#Print Incremented Date
#Date : 24/11/2021

from datetime import *

idate=input("Enter date in the format dd/mm/yyyy: ")
dd,mm,yyyy=idate.split("/")
dd=int(dd)
mm=int(mm)
yyyy=int/yyyy)
if mm in [1,3,5,7,8,10,12]:
    mv= 31
elif mm in [4,6,9,11]:
    mv= 30
elif (yyyy%4==0 and yyyy%100!=0 or yyyy%400==0):
    mv= 29
else:
    mv= 28

if mm < 1 or mm > 12 or dd < 1 or dd > mv :
    sys.exit("Invalid Date Entered, Recheck !")
elif dd==mv and mm!=12:
    dd= 1
    mm+= 1
elif dd== 31 and mm == 12:
    dd = 1
    mm = 1
    yyyy +=1
else:
    dd +=1

odate= date(yyyy,mm,dd)
fdate=odate.strftime("%d/%m/%Y")
print("Entered Date:" , idate)
print("Incremented Date: ",fdate)

```

Output Screenshots:

```

Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\7. if date valid print incremented.py
Enter date in the format dd/mm/yyyy: 27/5/2021
Entered Date: 27/5/2021
Incremented Date: 28/05/2021
>>>
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\7. if date valid print incremented.py
Enter date in the format dd/mm/yyyy: 31/12/2021
Entered Date: 31/12/2021
Incremented Date: 01/01/2022
>>>

```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\7. if date valid print incremented.py
Enter date in the format dd/mm/yyyy: 29/2/2020
Entered Date: 29/2/2020
Incremented Date: 01/03/2020
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\7. if date valid print incremented.py
Enter date in the format dd/mm/yyyy: 29/2/2019
SystemExit: Invalid Date Entered, Recheck !
>>>
```

Learning Outcome:

By performing this experiment, use of if-elif-else loop was understood. Use of Arithmetic Operators and Logical Operators was made clear. Appropriate data types were used to perform required operations and thus get required result. Use of appropriate assignment operators helped in achieving required result. This program helped in understanding the topics such as control flow statements, if-elif-else loop, use of inbuilt modules and functions, and in developing required logic for problem solving.

Experiment No. 6-B

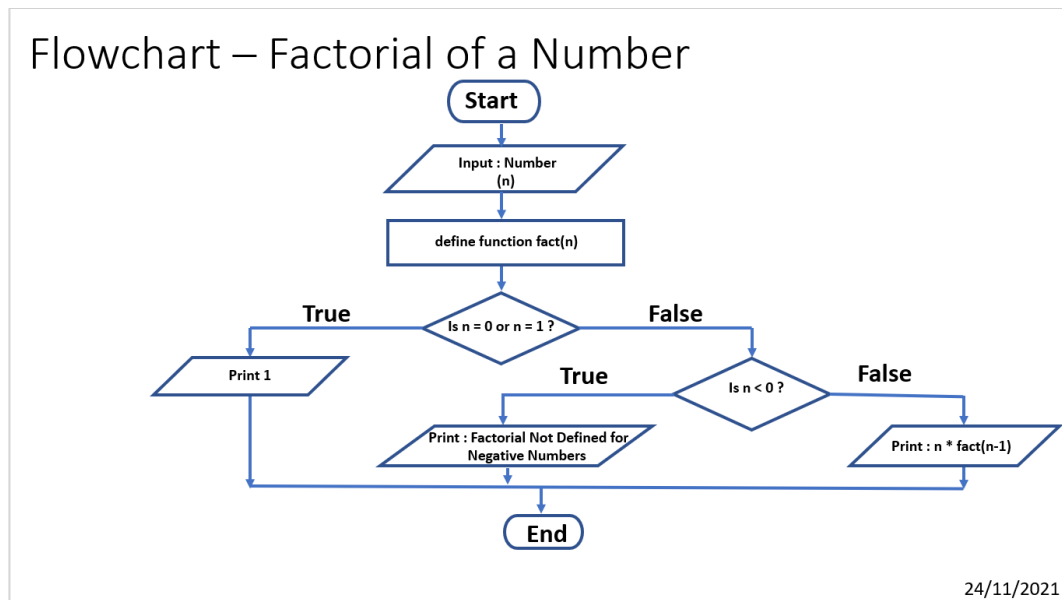
Title:

Write a Python Program to find the factorial of a number

Tool/Platform:

Microsoft Word / PowerPoint and Python IDLE

Flowchart:



Algorithm:

1. Initialise the program
2. Initialise variable “n”
3. Define a function **fact(n)** (which is recursive in nature)
4. Check if input number n is 1 or if it is 0
5. If true, print 1
6. If false, check if $n < 0$
7. If this is true, Print “**Factorial Not Defined for Negative Numbers**”
8. Else, return $n * \text{fact}(n-1)$

Source Code:

```
#Factorial of Number
# Date 24/11/2021
def fact(n):
    if n==1:
        return n
    else:
        return n * fact(n-1)

n=int(input("Enter Number: "))
if n<0:
    print("Factorial is not defined for Negative Numbers!")
elif n==0:
    print("Factorial = 1")
else:
    print("The factorial is:", fact(n))
```

```
#Factorial of Number
# Date 24/11/2021
def fact(n):
    if n==1:
        return n
    else:
        return n * fact(n-1)

n=int(input("Enter Number: "))
if n<0:
    print("Factorial is not defined for Negative Numbers!")
elif n==0:
    print("Factorial = 1")
else:
    print("The factorial is:", fact(n))
```

Output Screenshots:

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\5. Factorial of Number.py
Enter Number: 0
Factorial = 1
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\5. Factorial of Number.py
Enter Number: 1
The factorial is: 1
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\5. Factorial of Number.py
Enter Number: 5
The factorial is: 120
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\5. Factorial of Number.py
Enter Number: -2
Factorial is not defined for Negative Numbers!
>>>
```

Learning Outcome:

By performing this experiment, use of if-elif-else loop was understood. This program helped in understanding the topics such as control flow statements, if-elif-else loop, and in developing required logic for problem solving. By performing this experiment, use of user defined functions was understood and the how to define a recursive function was clear. Functions were used to structure the program and perform required operations to get desired results.

Experiment No. 7-A

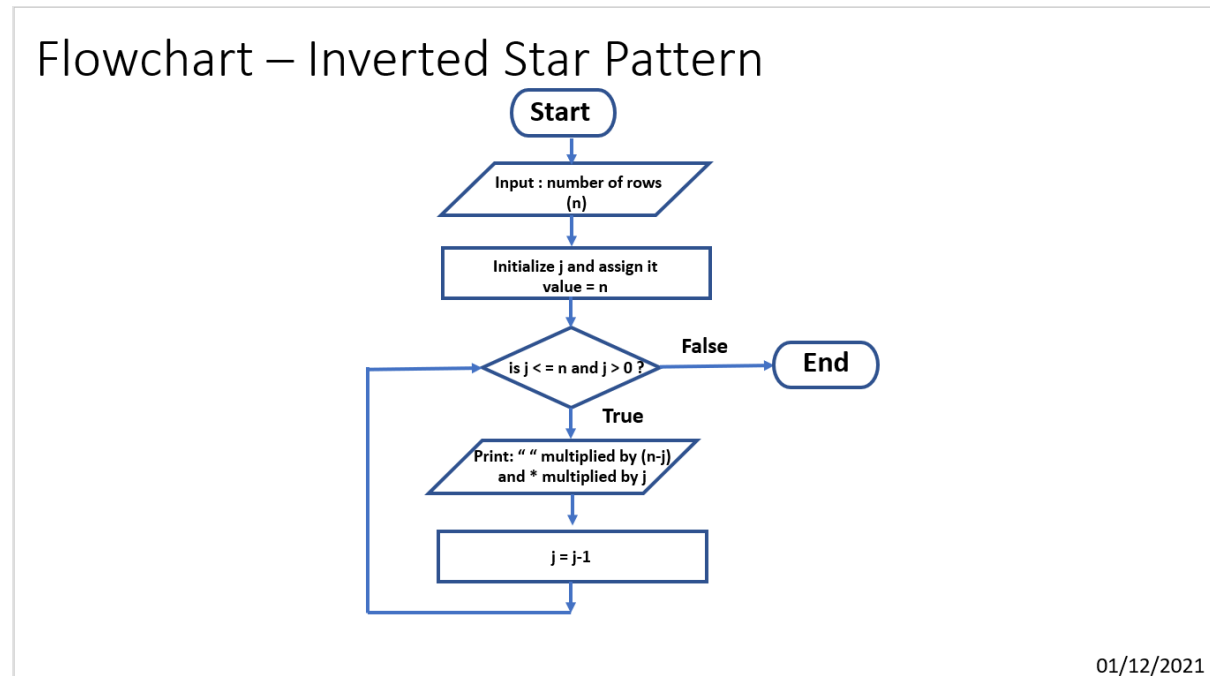
Title:

Write a Python Program to print an Inverted Star Pattern

Tool/Platform:

Microsoft Word / PowerPoint and Python IDLE

Flowchart:



Algorithm:

1. Initialise the program
2. Initialise variable “n”
3. Store input of number of rows from user in “n”
4. Using for loop iterate j in range n to 0, with decrement of 1 with each iteration
5. For each iteration multiply “ ” with n-j and * with j to manage spacing in rows
6. Print the concatenated result
7. End the program

Source Code:

```
#Inverted Star Pattern
# Date: 01/12/2021
n=int(input("Enter number of rows of the pattern: "))
print()
print("Printing Star Pattern:")
print()
for i in range(n):
    print(' '*(n-i-1)+'* '*(i+1))
print()
print("Printing Inverted Star Pattern:")
print()
for j in range(n,0,-1):
    print(' '*(n-j)+'* '*(j))

"""
n=int(input("Enter number of rows: "))
for i in range(0,n+1):
    print("* " * i)
print()
for j in range(n,0,-1):
    print("* " * j)
"""
```

```
#Inverted Star Pattern
# Date : 01/12/2021
n=int(input("Enter number of rows of the pattern: "))
print()
print("Printing Star Pattern:")
print()
for i in range(n):
    print(' '*(n-i-1)+'* '*(i+1))
print()
print("Printing Inverted Star Pattern:")
print()
for j in range(n,0,-1):
    print(' '*(n-j)+'* '*(j))

"""
n=int(input("Enter number of rows: "))
for i in range(0,n+1):
    print("* " * i)
print()
for j in range(n,0,-1):
    print("* " * j)
"""
```


Output Screenshots:

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\8. Inverted star pattern.py
Enter number of rows of the pattern: 10

Printing Star Pattern:

*
**
***
****
*****
*****
*****
*****
*****
*****

Printing Inverted Star Pattern:

*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

Learning Outcome:

By performing this experiment, it was understood that to solve a problem, one should look for the easiest and simplest approach possible. The syntax of for loop was understood and a basic program was written to solve the problem easily and efficiently.

Experiment No. 7-B

Title:

Write a Python Program to accept three digits and print all possible combinations of those digits

Tool/Platform:

Microsoft Word / PowerPoint and Python IDLE

Algorithm:

1. Initialise the program
2. Initialise variable "a", "b" and "c"
3. Store input of digits in "a", "b" and "c"
4. Create list "m" and input "a", "b" and "c" as its elements
5. Initialize empty list "n"
6. Using three for loops running from 0 to 3, to access the indexes of the elements of the list
7. The elements are printed if the values at the above indexes in the list are not equal
8. The above combination is appended into list n if not already existing in the list
9. End the program

Source Code:

```
# Input 3 digits, print max possible combination
```

```
# Date: 01/12/2021
```

```
import sys
```

```
a = int(input("Enter First Digit : "))
```

```
b = int(input("Enter Second Digit : "))
```

```
c = int(input("Enter Third Digit : "))
```

```
if a > 9 or b > 9 or c > 9:
```

```
    sys.exit("Invalid Input, Enter only Single Digits, Recheck !")
```

```
m=[a,b,c]
```

```
n=[]
```

```
print()
```

```
print("Printing all possible unique combinations using entered digits:", a,b,c)
```

```
print()
```

```
for i in range(0,3):
```

```
    for j in range(0,3):
```

```
        for k in range(0,3):
```

```
            if (i != j and j != k and k != i):
```

```

        x= [m[i] , m[j] , m[k]]
        if x not in n: # to find unique combinations
            n.append(x)
d=len(n)
for i in range(0,d):
    print(*n[i])

print()
print("Printed Number of Unique Combinations: ",d)

```

```

# Input 3 digits, print max possible combination
# Date: 01/12/2021

import sys

a = int(input("Enter First Digit : "))
b = int(input("Enter Second Digit : "))
c = int(input("Enter Third Digit : "))
if a > 9 or b > 9 or c > 9:
    sys.exit("Invalid Input, Enter only Single Digits, Recheck !")

m=[a,b,c]
n=[]
print()
print("Printing all possible unique combinations using entered digits:", a,b,c)
print()
for i in range(0,3):
    for j in range(0,3):
        for k in range(0,3):
            if (i != j and j != k and k != i):
                x= [m[i] , m[j] , m[k]]
                if x not in n: # to find unique combinations
                    n.append(x)
d=len(n)
for i in range(0,d):
    print(*n[i])

print()
print("Printed Number of Unique Combinations: ",d)

```

Output Screenshots:

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\9. max combination using 3 digits.py
Enter First Digit : 2
Enter Second Digit : 3
Enter Third Digit : 5

Printing all possible unique combinations using entered digits: 2 3 5

2 3 5
2 5 3
3 2 5
3 5 2
5 2 3
5 3 2

Printed Number of Unique Combinations: 6
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\9. max combination using 3 digits.py
Enter First Digit : 2
Enter Second Digit : 6
Enter Third Digit : 2

Printing all possible unique combinations using entered digits: 2 6 2

2 6 2
2 2 6
6 2 2

Printed Number of Unique Combinations: 3
>>>
```

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\gargm\Desktop\COLLEGE\Programming and Problem Solving Lab\9. max combination using 3 digits.py
Enter First Digit : 1
Enter Second Digit : 9
Enter Third Digit : 12
SystemExit: Invalid Input, Enter only Single Digits, Recheck !
>>>
```

Learning Outcome:

By performing this experiment, use of nested loops was understood. A code snippet was designed to filter out repeated output values from printing and thus only printing the necessary ones. It helped to understand the syntax of nested loops, and the use of lists to get required output in required format. The benefits of using lists and the operations/ functionalities they provide were made clear.