Name: Meet Raut
Batch: S21
Roll Number: 2201084

Python Lab

Experiment 1-B

AIM:- To study basic looping and if statements and implement calculator, age sorter and implement formulae.

THEORY:-

If Statements:

In Python, an if statement is used for conditional execution. It allows you to execute a block of code only if a certain condition is true. The general syntax of an if statement in Python is:

```
if condition:
        # code block to execute if condition is True
```

The condition can be any expression that evaluates to either True or False. If the condition is True, the code block following the if statement is executed. If the condition is False, the code block is skipped.

Branching:

Branching occurs when the flow of execution in a program diverges based on certain conditions. In Python, branching is commonly achieved using if, elif (short for "else if"), and else statements. Multiple conditions can be evaluated using a series of if-elif-else statements. The syntax looks like:

```
  if condition1:
    # code block to execute if condition1 is True
elif condition2:
    # code block to execute if condition1 is False and condition2 is True
else:
    # code block to execute if all conditions are False
```

Python evaluates the conditions in order. If a condition evaluates to True, the corresponding code block is executed, and the rest of the conditions are skipped. If none of the conditions are True, the code block under the else statement (if present) is executed.

Looping:

Looping is the process of repeatedly executing a block of code until a certain condition is met. Python provides two main loop constructs: the for loop and the while loop.

- For Loop: The for loop iterates over a sequence (such as a list, tuple, string, or range) and executes the code block for each item in the sequence. Here's the syntax:

```
for item in sequence:
   # code block to execute for each item in the sequence
```

- While Loop: The while loop repeats a block of code as long as a specified condition is True. Here's the syntax:

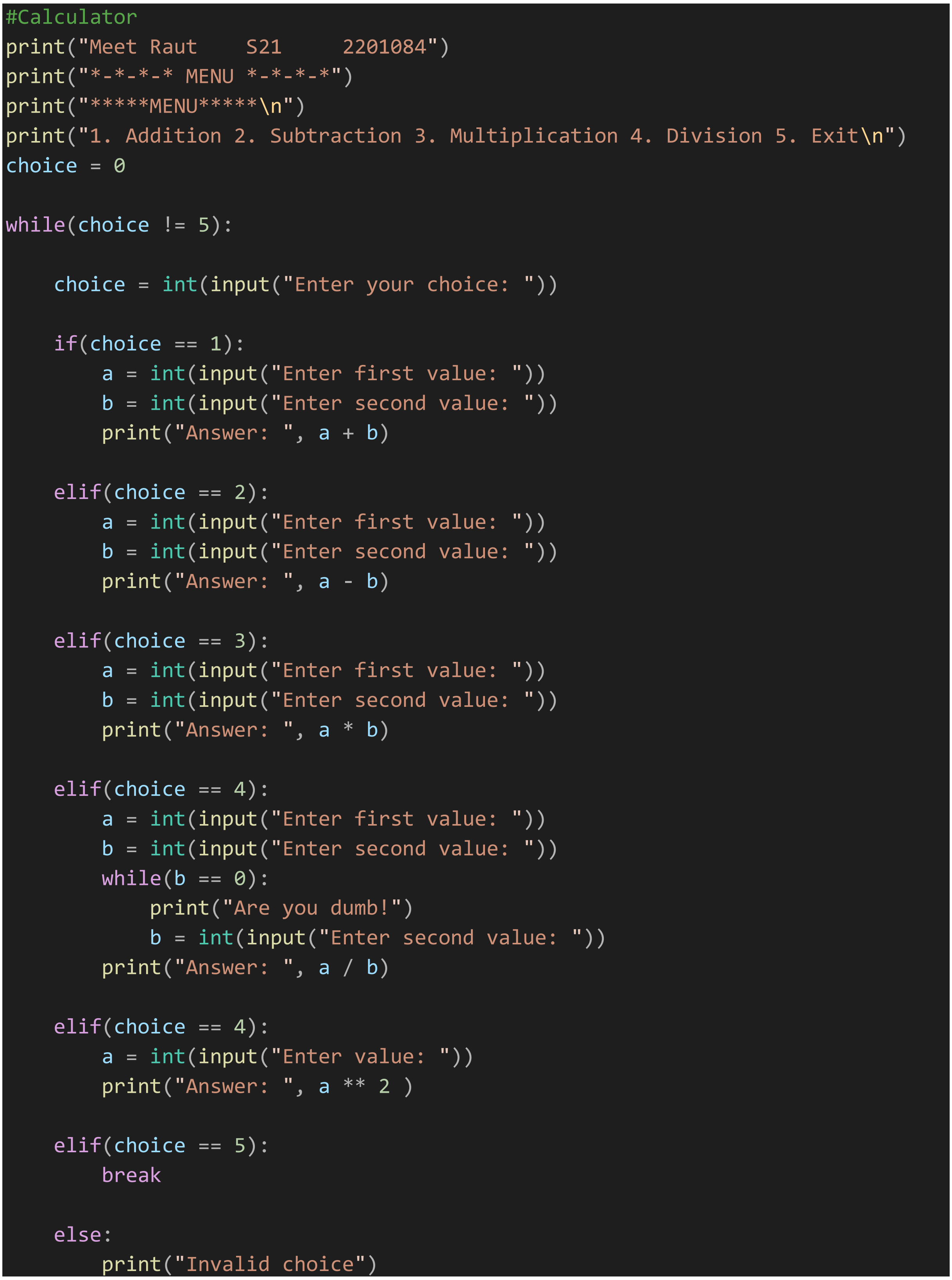
```
while condition:
   # code block to execute while the condition is True
```

It's important to ensure that the condition within a while loop eventually becomes False; otherwise, the loop will continue indefinitely, resulting in an infinite loop.

For this part of the experiment, we are implementing these control statements with three programs- a calculator, age sorter and an equation solver.

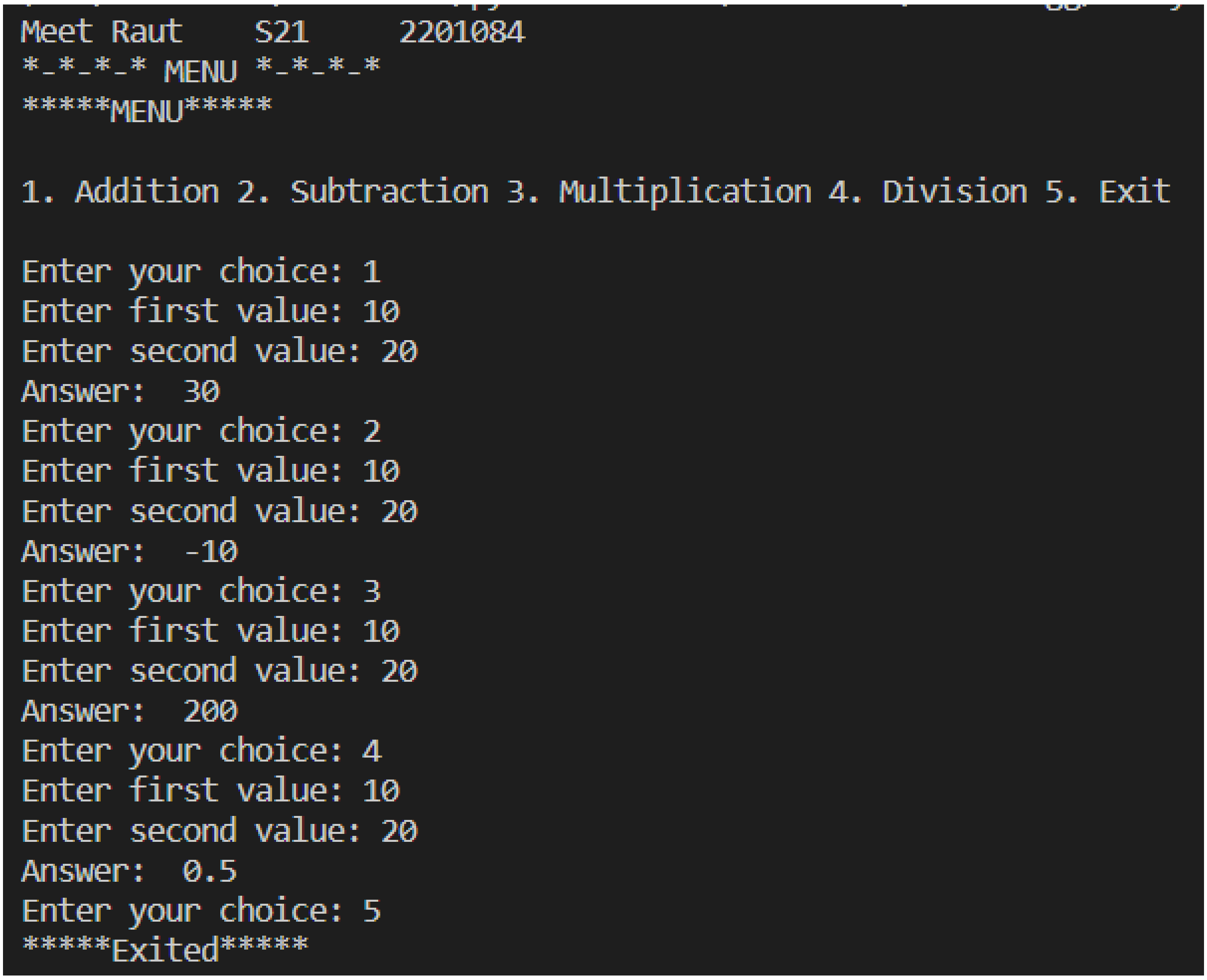
CODE:

1. Calculator



```
#Calculator
print("Meet Raut    S21     2201084")
print("*-*-*-* MENU *-*-*-*")
print("*****MENU*****\n")
print("1. Addition 2. Subtraction 3. Multiplication 4. Division 5. Exit\n")
choice = 0

while(choice != 5):

    choice = int(input("Enter your choice: "))

    if(choice == 1):
        a = int(input("Enter first value: "))
        b = int(input("Enter second value: "))
        print("Answer: ", a + b)

    elif(choice == 2):
        a = int(input("Enter first value: "))
        b = int(input("Enter second value: "))
        print("Answer: ", a - b)

    elif(choice == 3):
        a = int(input("Enter first value: "))
        b = int(input("Enter second value: "))
        print("Answer: ", a * b)

    elif(choice == 4):
        a = int(input("Enter first value: "))
        b = int(input("Enter second value: "))
        while(b == 0):
            print("Are you dumb!")
            b = int(input("Enter second value: "))
        print("Answer: ", a / b)

    elif(choice == 4):
        a = int(input("Enter value: "))
        print("Answer: ", a ** 2 )

    elif(choice == 5):
        break

    else:
        print("Invalid choice")
print("*****Exited*****")
```

OUTPUT:-



```
Meet Raut    S21     2201084
*_*_*_* MENU *_*_*_*
*****MENU*****

1. Addition 2. Subtraction 3. Multiplication 4. Division 5. Exit

Enter your choice: 1
Enter first value: 10
Enter second value: 20
Answer:  30
Enter your choice: 2
Enter first value: 10
Enter second value: 20
Answer:  -10
Enter your choice: 3
Enter first value: 10
Enter second value: 20
Answer:  200
Enter your choice: 4
Enter first value: 10
Enter second value: 20
Answer:  0.5
Enter your choice: 5
*****Exited*****
```

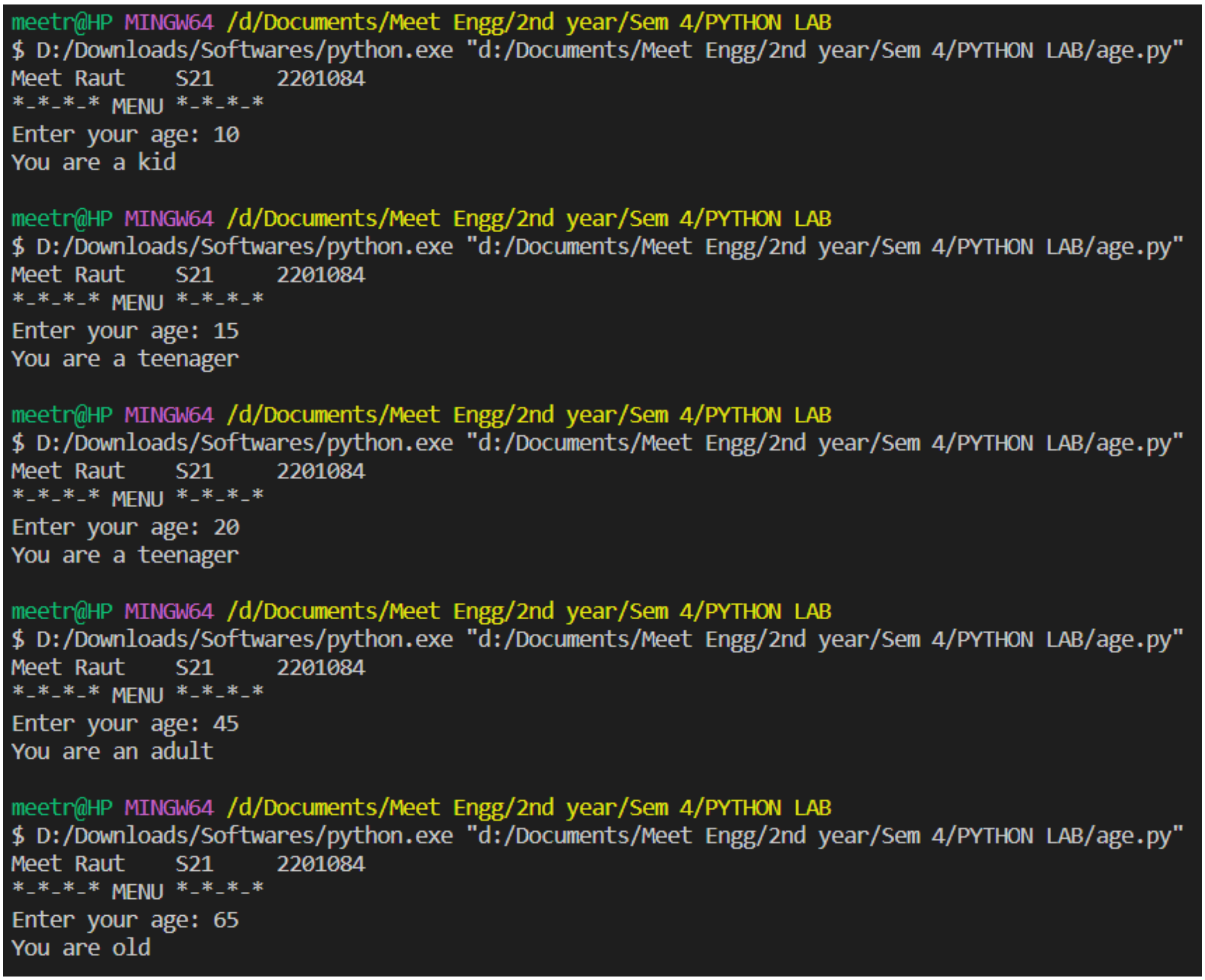
2. Age Sorter

CODE:

```
#Age
print("Meet Raut    S21     2201084")
print("*-*-*-* MENU *-*-*-*")
age = int(input("Enter your age: "))
if(age <= 0):
    print("INVALID")
elif(age <= 12):
    print("You are a kid")
```

```
elif(age <= 20):
    print("You are a teenager")
elif(age <= 50):
    print("You are an adult")
elif(age <= 100):
    print("You are old")
else:
    print("Invalid")
```

OUTPUT:

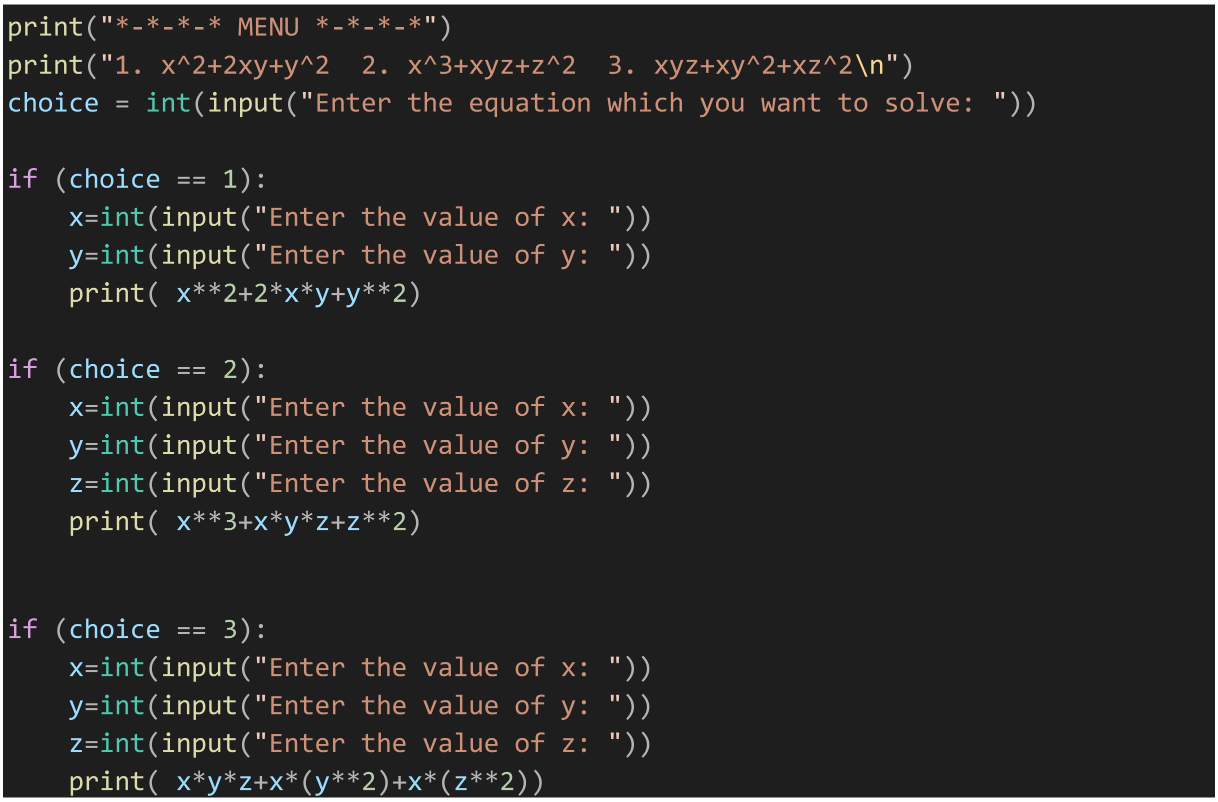


```
meetr@HP MINGW64 /d/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB
$ D:/Downloads/Softwares/python.exe "d:/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB/age.py"
Meet Raut    S21     2201084
*_*_*_* MENU *_*_*_*
Enter your age: 10
You are a kid

meetr@HP MINGW64 /d/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB
$ D:/Downloads/Softwares/python.exe "d:/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB/age.py"
Meet Raut    S21     2201084
*_*_*_* MENU *_*_*_*
Enter your age: 15
You are a teenager

meetr@HP MINGW64 /d/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB
$ D:/Downloads/Softwares/python.exe "d:/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB/age.py"
Meet Raut    S21     2201084
*_*_*_* MENU *_*_*_*
Enter your age: 20
You are a teenager

meetr@HP MINGW64 /d/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB
$ D:/Downloads/Softwares/python.exe "d:/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB/age.py"
Meet Raut    S21     2201084
*_*_*_* MENU *_*_*_*
Enter your age: 45
You are an adult

meetr@HP MINGW64 /d/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB
$ D:/Downloads/Softwares/python.exe "d:/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB/age.py"
Meet Raut    S21     2201084
*_*_*_* MENU *_*_*_*
Enter your age: 65
You are old
```

3. Equation Solver

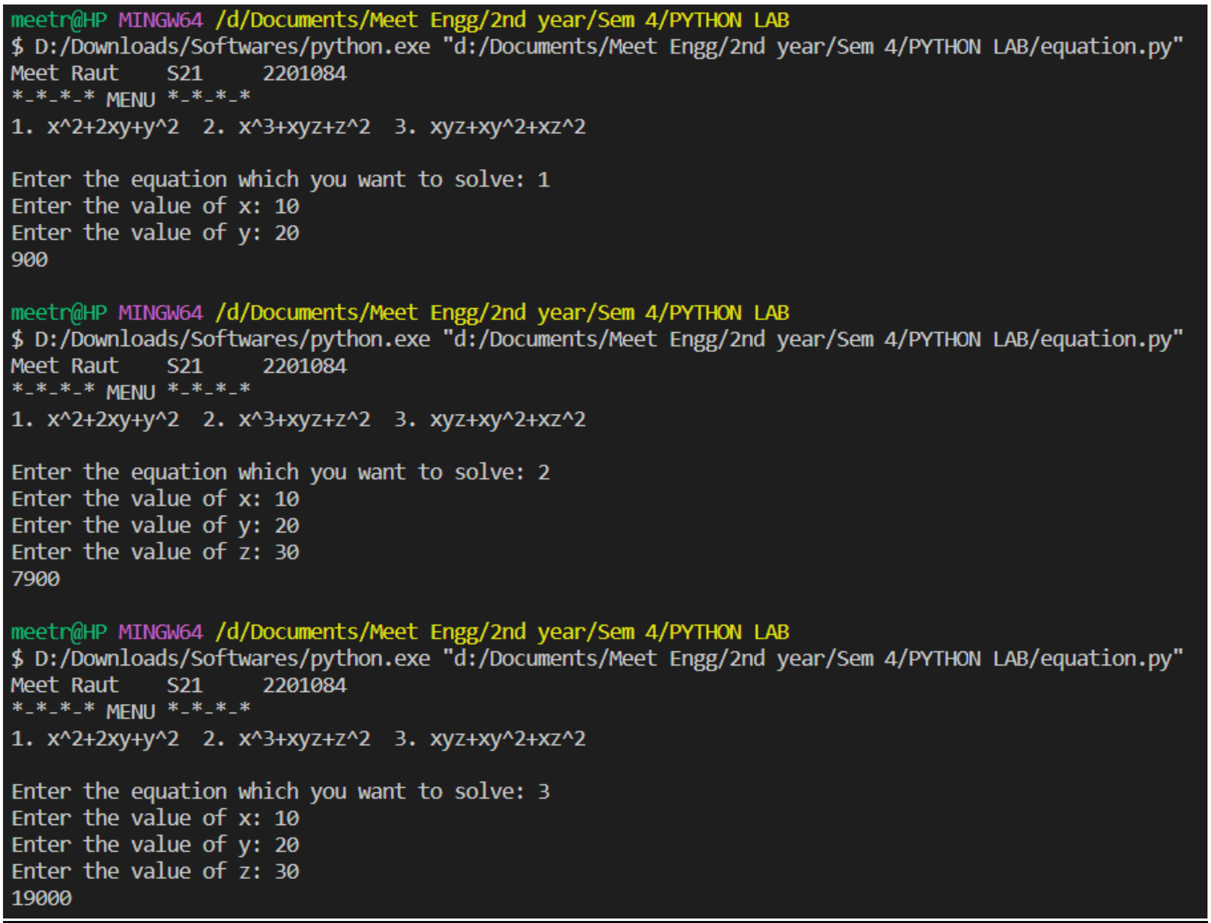
CODE:

```
#equation
print("Meet Raut    S21     2201084")
```



```
print("*-*-*-* MENU *-*-*-*")
print("1. x^2+2xy+y^2  2. x^3+xyz+z^2  3. xyz+xy^2+xz^2\n")
choice = int(input("Enter the equation which you want to solve: "))

if (choice == 1):
    x=int(input("Enter the value of x: "))
    y=int(input("Enter the value of y: "))
    print( x**2+2*x*y+y**2)

if (choice == 2):
    x=int(input("Enter the value of x: "))
    y=int(input("Enter the value of y: "))
    z=int(input("Enter the value of z: "))
    print( x**3+x*y*z+z**2)


if (choice == 3):
    x=int(input("Enter the value of x: "))
    y=int(input("Enter the value of y: "))
    z=int(input("Enter the value of z: "))
    print( x*y*z+x*(y**2)+x*(z**2))
```

OUTPUT:



```
meetr@HP MINGW64 /d/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB
$ D:/Downloads/Softwares/python.exe "d:/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB/equation.py"
Meet Raut    S21    2201084
*-*-*-* MENU *-*-*-*
1. x^2+2xy+y^2  2. x^3+xyz+z^2  3. xyz+xy^2+xz^2

Enter the equation which you want to solve: 1
Enter the value of x: 10
Enter the value of y: 20
900

meetr@HP MINGW64 /d/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB
$ D:/Downloads/Softwares/python.exe "d:/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB/equation.py"
Meet Raut    S21    2201084
*-*-*-* MENU *-*-*-*
1. x^2+2xy+y^2  2. x^3+xyz+z^2  3. xyz+xy^2+xz^2

Enter the equation which you want to solve: 2
Enter the value of x: 10
Enter the value of y: 20
Enter the value of z: 30
7900

meetr@HP MINGW64 /d/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB
$ D:/Downloads/Softwares/python.exe "d:/Documents/Meet Engg/2nd year/Sem 4/PYTHON LAB/equation.py"
Meet Raut    S21    2201084
*-*-*-* MENU *-*-*-*
1. x^2+2xy+y^2  2. x^3+xyz+z^2  3. xyz+xy^2+xz^2

Enter the equation which you want to solve: 3
Enter the value of x: 10
Enter the value of y: 20
Enter the value of z: 30
19000
```

CONCLUSION:

In conclusion, if statements, branching, and looping are fundamental control flow mechanisms in Python that allow you to make decisions and repeat actions based on conditions. Understanding these concepts is crucial for writing clear, efficient, and flexible code.