Practical No: 2

Aim: To develop a Python script using conditional statements (if, if-else) and loops (for, while, break, continue, range)

Theory:

Python provides control flow tools such as:

Conditional statements to make decisions:

`if` **Statement**

Executes a block if the condition is `True`.

Syntax:

```
if condition:

    # block of code
```

if-else Statement

Executes one block if the condition is `True`, else another block.

Syntax:

```
if condition:
    # block if true
else:
    # block if false
```

`if-elif-else Statement`

Checks multiple conditions in sequence.

Syntax:

```
if condition1:
    # block if condition1 is true
elif condition2:
    # block if condition2 is true
else:
    # block if none are true
```

Loops : Used to repeat a block of code multiple times.

`for` **Loop**

Iterates over a sequence (like list, string, or `range()`).

Syntax:

```
for variable in sequence:
        # block of code
```

Example with `range()`:

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

`while` **Loop**

Repeats the block as long as the condition is `True`.

Syntax:

```
while condition:
        # block of code
```

`break` **Statement**

Terminates the loop prematurely.

Syntax:

```
for i in range(10):
        if i == 5:
            break
```

`continue` **Statement**

Skips the current iteration and moves to the next one.

Syntax:

```
for i in range(10):
    if i % 2 == 0:
        continue
    print(i)
```

`range(start, stop, step)` **Function**

Used with `for` to generate a sequence of numbers.

Example:

```
    for i in range(0, 10, 2):
        print(
```

Program:

1. **Write a python program to find the largest of three numbers entered by the user.**

```
num1 = int(input("Enter first number: "))

num2 = int(input("Enter second number: "))

num3 = int(input("Enter third number: "))
```

```
if num1 >= num2 and num1 >= num3:

    largest = num1

elif num2 >= num1 and num2 >= num3:

    largest = num2

else:

    largest = num3


print("The largest number is:", largest)
```

OUTPUT:

```
PS C:\Users\ASUS\OneDrive\Desktop\python practicals\programs> pytho
ython practicals\programs\p21.py"
Enter first number: 54
Enter second number: 65
Enter third number: 25
The largest number is: 65
PS C:\Users\ASUS\OneDrive\Desktop\python practicals\programs>
```

2. **Electricity Bill Calculator**
   Create a Python program that takes the number of electricity units consumed as input and calculates the bill as per the following:

   ## Updated Charges:

   First 100 units → ₹6/unit
   Next 100 units (101–200) → ₹8/unit
   Beyond 200 units → ₹11/unit

```
units = int(input("Enter the number of electricity units consumed: "))
```

```
if units <= 100:

    bill = units * 6

elif units > 100 and units <= 200:

    bill = (100 * 6) + ((units - 100) * 8)

elif units > 200:
```

```
    bill = (100 * 6) + (100 * 8) + ((units - 200) * 11)

print("Total electricity bill: ₹", bill)
```

OUTPUT:

```
PS C:\Users\ASUS\OneDrive\Desktop\python practicals\programs> python -u "c:\
ython practicals\programs\p21.py"
Enter the number of electricity units consumed: 466
Total electricity bill: ₹ 4326
PS C:\Users\ASUS\OneDrive\Desktop\python practicals\programs>
```

3. **ATM PIN Verification**

Simulate an ATM system where a user has a maximum of 3 attempts to enter the correct PIN. On correct input, print "Access Granted", otherwise "Card Blocked" after 3 wrong attempts.

```
correct_pin = "1234"

attempts = 0



while attempts < 3:

    pin = input("Enter your ATM PIN: ")

    if pin == correct_pin:

        print("Access Granted")

        break

    else:

        attempts += 1

        print("Incorrect PIN")



if attempts == 3:

    print("Card Blocked")
```

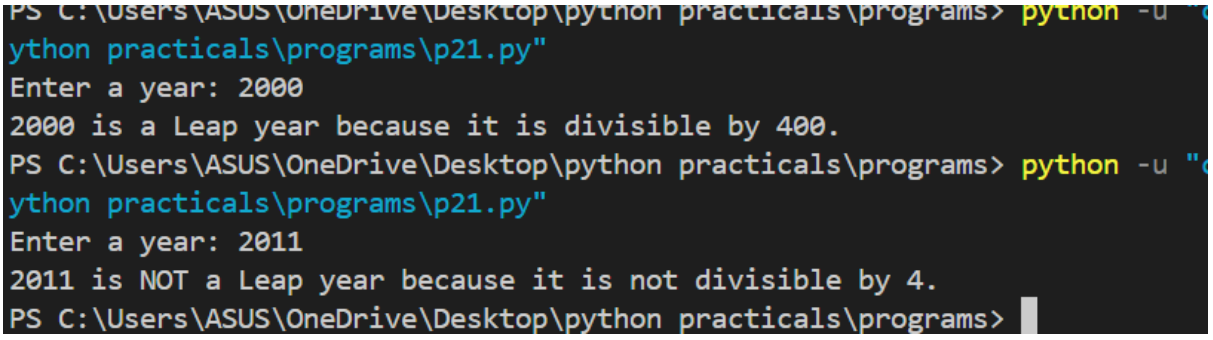
```
PS C:\Users\ASUS\OneDrive\Desktop\python practicals\programs> python -u "c:\Users\ASUS\OneDriv
ython practicals\programs\p21.py"
Enter your ATM PIN: 4565
Incorrect PIN
Enter your ATM PIN: 4665
Incorrect PIN
Enter your ATM PIN: 1234
Access Granted
```

4. **Leap year**

   Accept a year from the user and check if it is a leap year or not using the proper logic: Divisible by 4 but not 100, or divisible by 400.

```
year = int(input("Enter a year: "))
```

```
if year % 4 == 0:
    if year % 100 != 0:
        print(year, "is a Leap year because it is divisible by 4 but not by 100.")
    else:
        if year % 400 == 0:
            print(year, "is a Leap year because it is divisible by 400.")
        else:
            print(year, "is NOT a Leap year because it is divisible by 100 but not by 400.")
else:
    print(year, "is NOT a Leap year because it is not divisible by 4.")
```


PS C:\Users\ASUS\OneDrive\Desktop\python practicals\programs> python -u "
ython practicals\programs\p21.py"
Enter a year: 2000
2000 is a Leap year because it is divisible by 400.
PS C:\Users\ASUS\OneDrive\Desktop\python practicals\programs> python -u "
ython practicals\programs\p21.py"
Enter a year: 2011
2011 is NOT a Leap year because it is not divisible by 4.
PS C:\Users\ASUS\OneDrive\Desktop\python practicals\programs>