

# How to Use %s in Python

The %s operator is put where the string is to be specified

```
# declaring a string variable
name = "Geek"

# append a string within a string
print("Hey, %s!" % name)
```

Hey, Geek!

# How to Use %d in Python

```
# declaring numeric variables
num = 2021

# concatenating numeric value within string
print("%d is here!!" % num)
```

2021 is here!!

# randint() Function in Python

randint() is an inbuilt function of the random module in Python3.

```
# Python3 program explaining work  
# of randint() function
```

```
# imports random module  
import random
```

Random number between 0 and 10 is 0

```
# Generates a random number between  
# a given positive range  
r1 = random.randint(0, 10)  
print("Random number between 0 and 10 is % s" % (r1))
```

```
import random
r1 = random.randint(5, 15)
print("Random number between 5 and 15 is % s" % (r1))
r2 = random.randint(-10, -2)
print("Random number between -10 and -2 is % d" % (r2))
```

```
>>> == RESTART: C:/Users/warhlaingn/AppData/Local/Programs/Python/Python39-6/Python.exe
Random number between 5 and 15 is 11
Random number between -10 and -2 is -6
>>>
```

%s is used as a placeholder for string values into a formatted string.  
%d is used as a placeholder for numeric or decimal values

5. Create a variable called **secret\_num** and set the value to 50. Ask the user to enter a number. While their guess is not the same as the **secret\_num**, tell them if their guess is too low or too high, and ask them to have another guess. If they enter the same value as **secret\_num**, display the message 'Well done, you took [count] attempts'.

```
secret_num = 50
guess_num = int(input('Guess my secret number: '))
count = 1
while guess_num != secret_num:
    if guess_num < secret_num:
        print('Too Low')
    else:
        print('Too High')
    count = count+1
    guess_num=int(input('Have another guess: '))
print ('Well done, you took ' + str (count) + ' attempts.')
```

```
= RESTART: C:\Users\warhlaingn\AppData\l
Guess my secret number: 50
Well done, you took 1 attempts.
>>>
= RESTART: C:\Users\warhlaingn\AppData\l
Guess my secret number: 30
Too Low
Have another guess: 40
Too Low
Have another guess: 50
Well done, you took 3 attempts.
>>>
= RESTART: C:\Users\warhlaingn\AppData\l
Guess my secret number: 60
Too High
Have another guess: 55
Too High
Have another guess: 45
Too Low
Have another guess: 35
Too Low
Have another guess: 50
Well done, you took 5 attempts.
>>>
```

1. Improve the following Python program (Sample Answer for CE03 Q5) as in Listing 1 by:

- Randomise the `secret_num`. The random number generated must be within the range of 1 to 200.
- Add an iterative function using the while loop to ask the user for a decision to continue playing the game.

```
secret_num = 50
guess_num = int(input('Guess my secret number: '))
count = 1
while guess_num != secret_num:

    if guess_num < secret_num:
        print('Too Low')

    else:
        print('Too High')

    count = count+1

    guess_num=int(input('Have another guess: '))

print ('Well done, you took ' + str (count) + ' attempts.')
```

# How will you draw the flowchart?

- Randomise the `secret_num`. The random number generated must be within the range of 1 to 200.
- Same with the previous flowchart
- Add an iterative function using the while loop to ask the user for a decision to continue playing the game.

## # Question 1

```
import random
```

```
play_game = True
```

```
while play_game:
```

```
    secret_num = random.randint(1, 200)
```

```
    print(secret_num)
```

```
    guess_num = int(input('Guess my secret number: '))
```

```
    count = 1
```

```
    while guess_num != secret_num:
```

```
        if guess_num < secret_num:
```

```
            print('Too Low')
```

```
        else:
```

```
            print('Too High')
```

```
        count = count + 1
```

```
        guess_num = int(input('Have another guess: '))
```

```
    print('Well done, you took ' + str(count) + ' attempts.')
```

```
    cont_game = input("Do you want to continue? [y/n]: ").lower()
```

```
    while cont_game != 'n' and cont_game != 'y':
```

```
        cont_game = input("Do you want to continue? [y/n]: ").lower()
```

```
    if cont_game == 'n':
```

```
        play_game = False
```

1

Randomise the `secret_num`.  
The random number generated must be within the range of 1 to 200.

2

Not changing

3

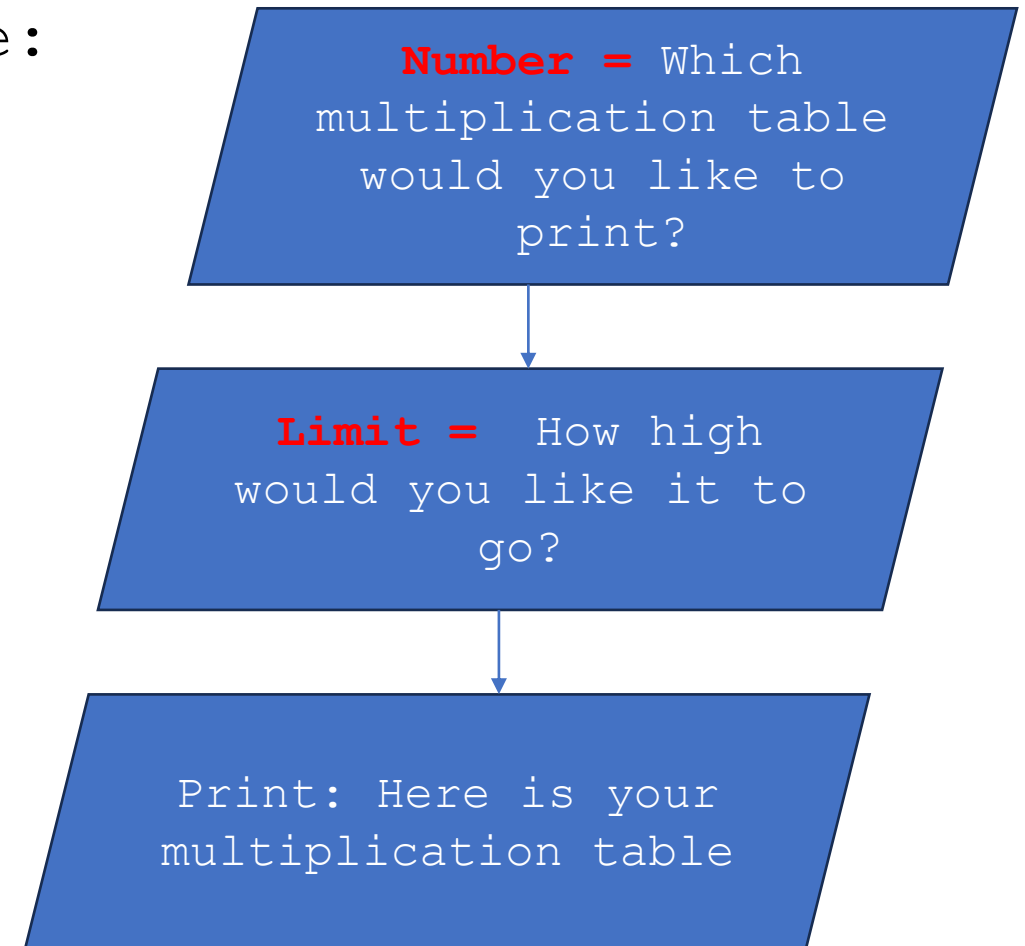
Add an iterative function using the while loop to ask the user for a decision to continue playing the game.

2. Write a Python program using the while loop to generate a multiplication table that will have the following outputs.

```
Which multiplication table would you like to print? 8
How high would you like it to go? 10
```

Here is your multiplication table:

```
8 times 1  =   8
8 times 2  =  16
8 times 3  =  24
8 times 4  =  32
8 times 5  =  40
8 times 6  =  48
8 times 7  =  56
8 times 8  =  64
8 times 9  =  72
8 times 10 =  80
```





```
# Input the number for which multiplication table needs to be generated
num = int(input("Which multiplication table would you like to print? "))

# Input the range for the multiplication table
range_limit = int(input("How high would you like it to go? "))

# Initialize a counter
i = 1

print("\nHere is your multiplication table:")

# Loop through the range and print the multiplication table
while i <= range_limit:
    result = num * i
    print(f"{num} times {i} = {result}")
    i += 1
```

```
8 times 1 = 8
8 times 2 = 16
8 times 3 = 24
8 times 4 = 32
8 times 5 = 40
8 times 6 = 48
8 times 7 = 56
8 times 8 = 64
8 times 9 = 72
8 times 10 = 80
```

3. Write a Python program using the while loop to print the following pattern.

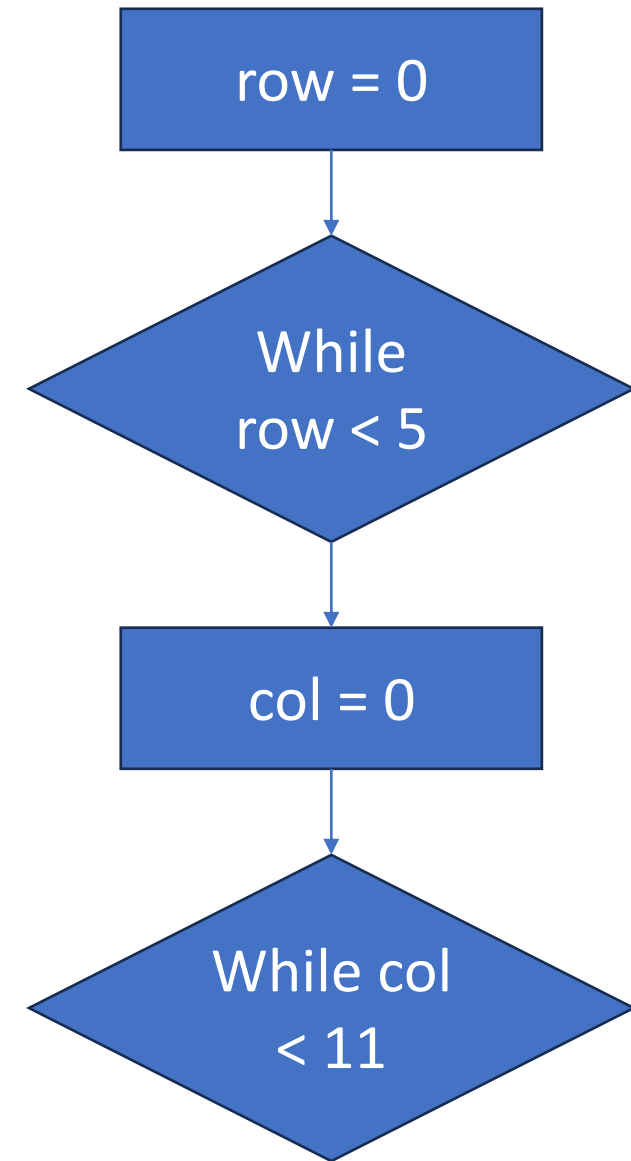
```
10101010101
10101010101
10101010101
10101010101
10101010101
```

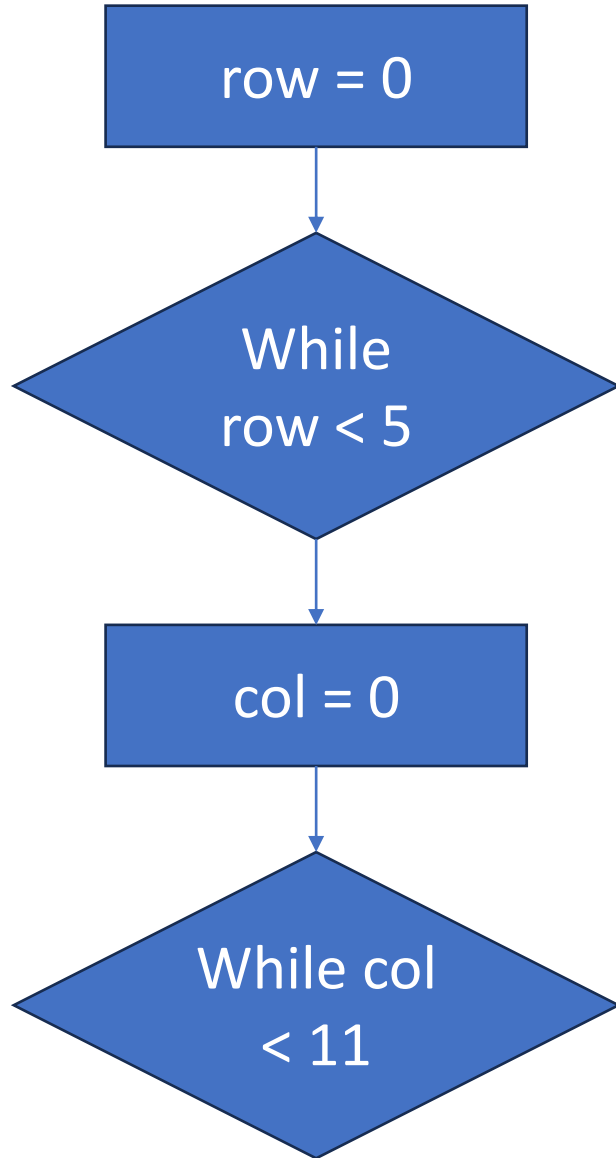
Do not use while loop to print the pattern ( '10101010101' ) for 5 times.

The program should print the pattern **character-by-character**.

Row = 0 to 4 (5 rows)  
Columns = 0 to 10 (11 columns)

$C_0$											$C_{10}$	
1	0	1	0	1	0	1	0	1	0	1		$R_0$
1	0	1	0	1	0	1	0	1	0	1		$R_1$
1	0	1	0	1	0	1	0	1	0	1		$R_2$
1	0	1	0	1	0	1	0	1	0	1		$R_3$
1	0	1	0	1	0	1	0	1	0	1		





```
row = 0
while row < 5:
    col = 0
    while col < 11:

        col = col + 1
    print ()
    row += 1
```

# 101010101

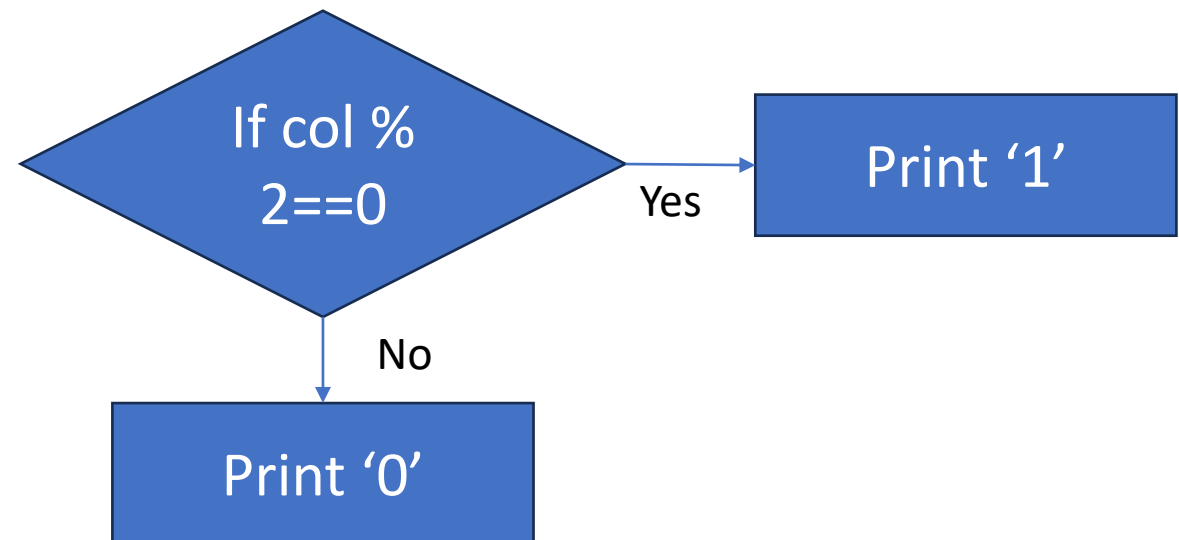
If  $\text{col} \% 2 == 0$

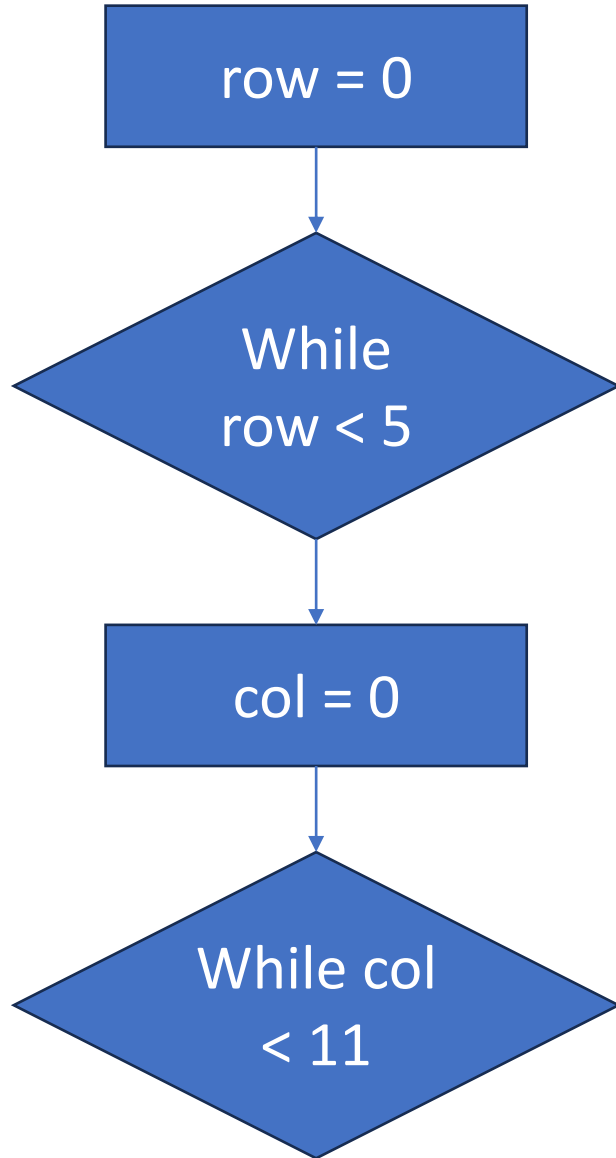
$0\%2=0$ ,	print 1	= 1
$1\%2=1$ ,	else print 0	= 0
$2\%2=0$ ,	print 1	= 1
$3\%2=1$ ,	else print 0	= 0
$4\%2=0$ ,	print 1	= 1
$5\%2=1$ ,	else print 0	= 0
$6\%2=0$ ,	print 1	= 1
$7\%2=1$ ,	else print 0	= 0
$8\%2=0$ ,	print 1	= 1
$9\%2=1$ ,	else print 0	= 0
$10\%2=0$ ,	print 1	= 1

❖ We want to print 1 and 0 alternately

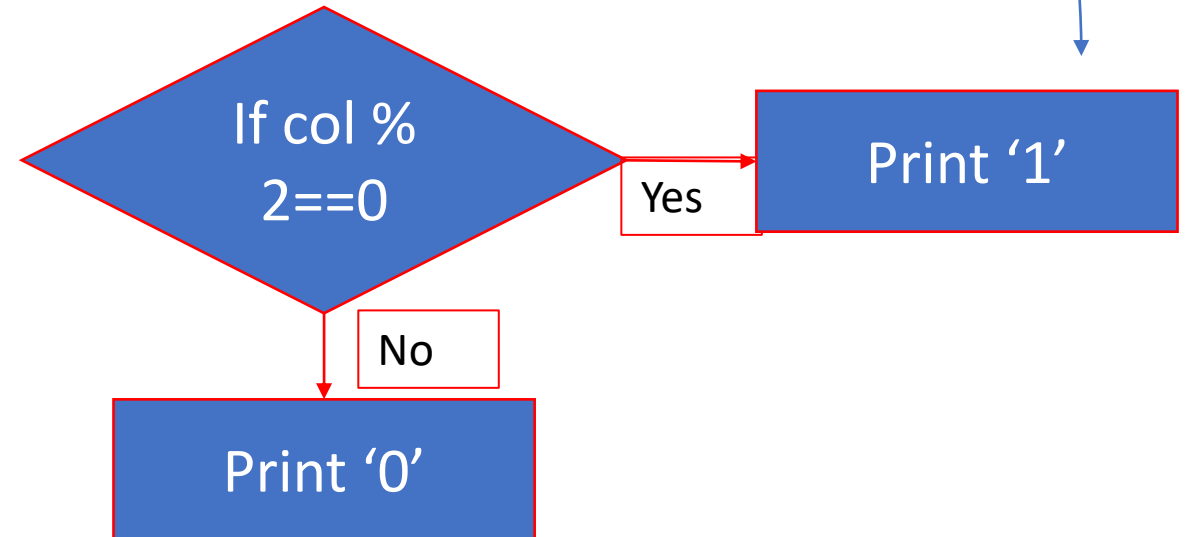
❖ We will use % operator

❖ 0 and 1 = two conditions, so we will use if ...else statements





```
row = 0
while row < 5:
    col = 0
    while col < 11:
        if col % 2 == 0:
            print ('1', end = '')
        else:
            print ('0', end = ' ')
        col = col + 1
    print ()
    row += 1
```



```
row = 0
while row < 5:
    col = 0
    while col < 11:
        if col % 2 == 0:
            print ('1', end = ' ')
        else:
            print ('0', end = ' ')
        col = col + 1
    print ()
    row = row + 1
```

Not going new line

going new line

```
>>>
→ = RESTART: C:\Users\warhlaingn\
10101010101
10101010101
10101010101
10101010101
10101010101
>>>
```

4. Write a temperature conversion program between degree Celsius and degree Fahrenheit, according to user selection. The following is the formula:

$$\text{Celsius} = (\text{Fahrenheit} - 32) * 5/9$$

$$\text{Fahrenheit} = (\text{Celsius} * 9/5) + 32$$

- The temperature conversion program is expected to convert a range of temperature measurements from a minimum value to a maximum value.
- The program will only execute if the minimum temperature is smaller than or equal to the maximum temperature. Else, an error message should be prompted to the user and the program restart.
- The program will only execute when the menu selection is valid. Else, an error message should be prompted to the user and the program restart.
- Write an iteration check if the user would like to run the program again.



Start

Input Selection (1 or 2) -----> Is Selection Valid? -----> Invalid Selection

No

Input Temperatures --> Are Temperatures Valid? --> Invalid Input

No

No

Invalid Input

Perform Conversion

Perform Conversion

Output Results

End

Do you want to run the program again? (Yes/No) --> Yes --> Start

No

End

## Screen output examples:

```
Temperature Conversion Programme.  
1.Convert Celsius to Fahrenheit.  
1.Convert Fahrenheit to Celsius.  
Enter your selection, 1 or 2: 1  
Celsius (C) to Fahrenheit (F) Conversion  
Enter temperature in integer values only.  
Enter minimum temperature: 20  
Enter maximum temperature: 25  
20C = 68.0F  
21C = 69.8F  
22C = 71.6F  
23C = 73.4F  
24C = 75.2F  
25C = 77.0F Conversion Done!  
Do you want to run the program again? [Y/N]: Y
```

```

play_again = 'Y'

while play_again == 'Y':

    print ('Temperature Conversion Programme.')
    print ('[1] Convert Celsius to Fahrenheit.')
    print ('[2] Convert Fahrenheit to Celsius.')

    selection = int (input('Enter your selection, 1 or 2: '))

    if selection == 1:
        print ('Celsius (C) to Fahrenheit (F) Conversion')
        print ('Enter temperature in interger values only.')
        temp_min = int(input('Enter minimum temperature: '))
        temp_max = int(input('Enter maximum temperature: '))

        if temp_min <= temp_max:
            temp_c = temp_min
            while temp_c <= temp_max:
                temp_f = (temp_c * 9/5) + 32
                print(f'{temp_c:>5.1f}C = {temp_f:5.1f}F')
                temp_c = temp_c + 1
            print ('Conversion Done.')
        else:
            print ('Error: Invalid Input!')

    elif selection == 2:
        print ('Fahrenheit (F) to Celsius (C) Conversion')
        print ('Enter temperature in interger values only.')
        temp_min = int(input('Enter minimum temperature: '))
        temp_max = int(input('Enter maximum temperature: '))

        if temp_min <= temp_max:
            temp_f = temp_min
            while temp_f <= temp_max:
                temp_c = (temp_f-32)*5/9
                print(f'{temp_f:>5.1f}F = {temp_c:5.1f}C')
                temp_f = temp_f+1
            print ('Conversion Done.')
        else:
            print ('Error: Invalid Input!')

    else:
        print ('Error: Invalid Selection!')

    play_again = input('Do you want to run the program again? [Y/N]: ').upper()
    while play_again != 'N' and play_again != 'Y':
        play_again = input('Do you want to run the program again? [Y/N]: ').upper()

print ('Program Terminated.')

```

```
play_again = 'Y'

while play_again == 'Y':

    print ('Temperature Conversion Programme.')
    print ('[1] Convert Celsius to Fahrenheit.')
    print ('[2] Convert Fahrenheit to Celsius.')

    selection = int (input('Enter your selection, 1 or 2: '))
```

### Screen output examples:

```
Temperature Conversion Programme.
1.Convert Celsius to Fahrenheit.
1.Convert Fahrenheit to Celsius.
Enter your selection, 1 or 2: 1
```

```
if selection == 1:
    print('Celsius (C) to Fahrenheit (F) Conversion')
    print('Enter temperature in integer values only.')
    temp_min = int(input('Enter minimum temperature: '))
    temp_max = int(input('Enter maximum temperature: '))
```

```
    if temp_min <= temp_max:
        temp_c = temp_min
        while temp_c <= temp_max:
            temp_f = (temp_c * 9/5) + 32
            print(f'{temp_c:>5.1f}C = {temp_f:>5.1f}F')
            temp_c = temp_c + 1
        print('Conversion Done.')
    else:
        print('ERROR: Invalid Input!')
```

The program will only execute if the minimum temperature is smaller than or equal to the maximum temperature.

```
Celsius (C) to Fahrenheit (F) Conversion
Enter temperature in integer values only.
Enter minimum temperature: 20
Enter maximum temperature: 25
```

Temperature Conversion Programme.

1.Convert Celsius to Fahrenheit.

1.Convert Fahrenheit to Celsius. Enter your selection, 1 or 2: 3

ERROR: Invalid Selection!

Do you want to run the program again? [Y/N]: Y

Temperature Conversion Programme.

1.Convert Celsius to Fahrenheit.

1.Convert Fahrenheit to Celsius.

Enter your selection, 1 or 2: 2

Fahrenheit (F) to Celsius (C) Conversion Enter temperature in integer values only.

Enter minimum temperature: 50

Enter maximum temperature: 30

ERROR: Invalid Input!

Do you want to run the program again? [Y/N]: Y

```
elif selection == 2:
    print('Fahrenheit (F) to Celsius (C) Conversion')
    print('Enter temperature in integer values only.')
    temp_min = int(input('Enter minimum temperature: '))
    temp_max = int(input('Enter maximum temperature: '))

    if temp_min <= temp_max:
        temp_f = temp_min
        while temp_f <= temp_max:
            temp_c = (temp_f - 32) * 5/9
            print('{temp_f:>5.1f}F = {temp_c:>5.1f}C')
            temp_f = temp_f + 1
            print('Conversion Done.')
        else:
            print('ERROR: Invalid Input!')
    else:
        print('ERROR: Invalid Selection!')
```

```
play_again = input('Do you want to run the program again? [Y/N]: ').upper()

while play_again != 'N' and play_again != 'Y':
    play_again = input('Do you want to run the program again? [Y/N]: ').upper()

print ('Program Terminated.')
```



```
print(f'{temp_c:>5.1f}C = {temp_f:5.1f}F')
```

`f'{temp_c:>5.1f}C = {temp_f:5.1f}F':`

This is a formatted string literal (also known as f-string), indicated by the **f prefix** before the string. It allows you to embed expressions within curly braces `{}` within the string.

`{temp_c:>5.1f}`: This part of the f-string formats the value `temp_c`, which likely represents a temperature in Celsius. Here's what each component does:

`:>5.1f`: This is the formatting specifier.

`>`: This is the alignment specifier, indicating that the value should be **right-aligned**.

`5`: This is the minimum width specifier. It specifies that the resulting string should be **at least 5 characters wide**. If the value `temp_c` is less than 5 characters wide, it will be padded with spaces to meet this width.

`.1f`: This is the format specifier for floating-point numbers. It indicates that the number should be formatted with **one digit after the decimal point**.