Title: Mastering Control Structures, Loops, and Operators in Python

Objective:

The purpose of this assignment is to enhance your proficiency in Python programming through challenging tasks. You will develop skills in control flow statements (if-elif-else), loops (for, while), nested loops, pattern programming, and more. The assignment also focuses on handling infinite loops, understanding loop control mechanisms (break, continue, pass), and using operators and escape characters. Additionally, tasks will challenge your understanding of input/output operations and type casting.

Task 1: Write a Python program to find the largest of three numbers.

In [ ]:

```python
# Taking input from users
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))
c = int(input("Enter the third number: "))

# using if-elif-else to find the largest number.
if a>=b and a>=c:
  print("The largest number is:",a)
elif b>=a and c>=a:
  print("The largest number is:",b)
else:
  print("The largest number is:",c)
```

```
Enter the first number: 12
Enter the second number: 14
Enter the third number: 9
The largest number is: 9
```

Task 2: Create a program that checks if a number is even or odd.

In [ ]:

```python
# Taking input from users
number = int(input("Enter The number: "))

# Using if-else to check even or odd.
if number % 2 == 0:
  print("Number is Even:", number)
else:
  print("Number is Odd:", number)
```

```
Enter The number: 9
Number is Odd: 9
```

Task 3: Write a program that prints numbers from 1 to 100 but skips numbers divisible by 7.

In [ ]:

```python
for i in range(1,101):
  if i % 7 == 0:
    continue  # skip those numbers who divisible by 7
  print(i,end=',')
```

```
1,2,3,4,5,6,8,9,10,11,12,13,15,16,17,18,19,20,22,23,24,25,26,27,29,30,31,32,33,3
4,36,37,38,39,40,41,43,44,45,46,47,48,50,51,52,53,54,55,57,58,59,60,61,62,64,65,6
6,67,68,69,71,72,73,74,75,76,78,79,80,81,82,83,85,86,87,88,89,90,92,93,94,95,96,9
7,99,100,
```

Task 4: Write a program to print the multiplication table of a given number.

In [ ]:

```
number = int(input("Enter the number: "))

for i in range(1,11):
  print(number, "X", i, "=", number*i)
```

```
Enter the number: 8
8 X 1 = 8
8 X 2 = 16
8 X 3 = 24
8 X 4 = 32
8 X 5 = 40
8 X 6 = 48
8 X 7 = 56
8 X 8 = 64
8 X 9 = 72
8 X 10 = 80
```

Task 5: Write a program that uses a nested loop to print a right-angled triangle pattern of stars based on user input.

In [ ]:

```
rows = int(input("Enter the number of rows:"))

for i in range(1,rows+1):   # outer loop rows
  for j in range(i):        #inner loop
    print("*",end='')
  print()
```

```
Enter the number of rows:4
*
**
***
****
```

Task 6: Create a Python program that prints the following pattern using nested loops:

```
1
12
123
1234
```

In [ ]:

```
for i in range(1,5):
  for j in range(1,i+1):
    print(j,end='')
  print()
```

```
1
12
123
1234
```

Task 7: Create a Python program to check if a character entered by the user is a vowel or consonant.

In [ ]:

```
char = input("Enter the Character: ").lower()

if len(char) == 1 and char.isalpha():
  if char in 'aeiou':
    print(f"{char} is a vowel.")
  else:
    print(f"{char} is a consonant.")
else:
  print("Invalid input. Please enter a single alphabet character.")
```

```
Enter the Character: B
b is a consonant.
```

Task 8. Write a Python program that generates a diamond pattern of stars For n = 5, the output should look like:

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

In [10]:

```
n = 5
for i in range(1,n+1):
  stars = '*' *(2*i-1)
  print(stars)

for i in range(n-1,0,-1):
  stars = '*' *(2*i-1)
  print(stars)
```

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

Task 9: Write a Python program to reverse the digits of a given number.

In [6]:

```
number = int(input("Enter the number: "))
reverse_no = 0

while number > 0:
  digit = number % 10
  reverse_no = reverse_no * 10 + digit
  number = number // 10
```

```
print(reverse_no)
```

```
Enter the number: 12345
54321
```

Task 10. Create a Python program that generates the following pyramid pattern:

```
1
121
12321
1234321
123454321
```

In [9]:

```
n = 5

for i in range(1,n+1):
  for j in range(1,i+1):
    print(j,end='')

  for j in range(i-1,0,-1):
    print(j,end='')
  print()
```

```
1
121
12321
1234321
123454321
```

In [ ]: