```
# Loops
# Loops are used to execute a set of statements
# repeatedly until a given condition is False

# 2 Loops in Python
#1. for loop
#2. while loop

# range function
# range is an in-built class.
"""
Syntax:
range([start,]stop[,step])
range will generate a range object
range will produce a sequence of values from (start) till (stop - 1)
with a difference of step between each value.
1. start parameter is optional.
   Default value for start parameter is 0
2. stop is mandatory.
   stop value is always excluded.
3. step parameter is optional
   Default value for step is 1
4. range will also accept negative numbers.
"""


# print(list(range()))
print(range(10))
print(list(range(10)))
print(list(range(5)))
print(list(range(10, 20)))
print(list(range(10, 20, 2)))
print(list(range(10, 20, 3)))

print(list(range(-20, -10))) # [-20, -19, -18, -17, -16, -15, -14, -13, -12, -11]
print(list(range(-10, -20, -1))) # [-10, -11, -12, -13, -14, -15, -16, -17, -18,
-19]

print(list(range(-10, -20))) # []
print(list(range(-20, -40, 2))) # []
print(list(range(40, 50, -1))) # []

print(list(range(20, 10, -1))) # [20, 19, 18, 17, 16, 15, 14, 13, 12, 11]
print(list(range(100, 50, -5))) # [100, 95, 90, 85, 80, 75, 70, 65, 60, 55]
print(list(range(-30, -10, 2))) # [-30, -28, -26, -24, -22, -20, -18, -16, -14,
-12]
print(list(range(-30, -50, -2))) # [-30, -32, -34, -36, -38, -40, -42, -44, -46,
-48]

# For Loop
# Syntax
# for <variable_name> in range([start,]stop[, step]):
#     # Body

# Example:
for i in range(5):
    print(i, end=' ')
print("Line 55")
```

```
for i in range(-10, -20):
    print("Hi")


# While Loop
# Syntax:
# Initialization - Assigning value to a variable
# Initialization will execute only once
# Loop will iterate until the given condition is False
"""
Initialization
while <condition>:
    # Body
    Update
"""

# Example
i = 0
while i < 5:
    print(i, end=' ')
    i += 1

# Infinite Loops:
# Missing condition inside a loop leads the loop to
# execute continuously without stopping.
"""
i = 0
while i < 5:
    print(i, end=' ')
"""



# Loop Control Statements
"""
1. Break - When a break statement is  encountered,
           control comes out of the loop.
           All the next iterations will be skipped.
2. Continue - When a continue statement is encountered,
              The current iteration will be skipped and
              goes to the next iteration.
Iteration - Execution of the loop body
These are used with Conditional Statements.
"""
# Break
for i in range(10):
    if i == 5:
        break
    print("i value is ", i)
print("Outside the Loop")


i = 0
while i < 10:
    if i == 5:
        break
    print(i, end=' ')
    i += 1
print("Outside the Loop")
```

```
# Continue
for i in range(10):
    if i == 5:
        continue
    print("i value is ", i)
print("Outside the Loop")


i = 0
while i < 10:
    i += 1
    if i == 5:
        continue
    print(i, end=' ')
print("Outside the Loop")


print("----------------------------")
for i in range(10):
    break
    print(i, end=' ')
print("Outside the Loop")

for i in range(10):
    continue
    print(i, end=' ')
print("Outside the Loop")


# Else Block with Loops:
"""
1. Else is optional.
2. Code inside the Else block will be executed only
   after the loop has successfully completed all the iterations.
3. If the loop encounters a break statement,
   the else block will be skipped.

# Syntax:
# else with for loop
for <variable_name> in range([start,]stop[,step]):
    # Body
else:
    # Body

# else with while loop
<initialization>
while <condition>:
    # Body
    # Update
else:
    # Body
"""

"""
for i in range(10):
    print(i, end=' ')
else:
```

```
    print("Loop has finished execution.")


i = 0
while i < 5:
    print(i, end=' ')
    i += 1
else:
    print("Loop Execution completed..")

# else block with break and continue statements
# else with break statement
for i in range(10):
    print(i, end=' ')
    if i == 5:
        break
else:
    print("Loop has finished execution.")


print()
i = 0
while i < 5:
    print(i, end=' ')
    i += 1
    if i == 5:
        break
else:
    print("Loop Execution completed..")


# else with continue statement
for i in range(10):
    print(i, end=' ')
    if i == 5:
        continue
else:
    print("Loop has finished execution.")


print()
i = 0
while i < 5:
    print(i, end=' ')
    i += 1
    if i == 5:
        continue
else:
    print("Loop Execution completed..")


# print if a number is in between the given range of 2 numbers.
# Input: n1 = 10, n2 = 30, x = 15
n1 = int(input("Enter N1: "))
n2 = int(input("Enter N2: "))
x = int(input("Enter X: "))
for i in range(n1, n2 + 1):
    if i == x:
        print(f"{x} is in between {n1} and {n2}")
```

```
        break
else:
    print(f"{x} is not in between {n1} and {n2}")
"""


# When to use which loop:
"""
When we know the number of iterations, we use for loop.
If the iterations are unknown, we use while loop.
"""


# Nested Loops
# For each iteration of outer loop, the inner loop will
# execute from start to finish.
m = int(input("Enter m: "))
n = int(input("Enter n: "))
for i in range(m):
    for j in range(n):
        print(f'{i}-{j}', end=' ')
    print()
# Time Complexity - O(m * n)


for i in range(m):
    print(i)
for j in range(n):
    print(j)
# Time Complexity - O (m + n)
```