QUIZ - Loops

Q 1:

Define a function that draw a square of stars on the screen.

The function name will be **sqaure_of_stars_with_while**.

The parameter will be number of stars in one side of square.

Hints:

- ask for number of stars from the user
- use while loop

```
# Q 1:

# ----- your solution here -----
def square_of_stars_with_while(n):
    i = 0
    while i < n:
        j = 0
        while j < n:
            print("*", end=" ")
            j += 1
        print()
        i += 1
# call the function you defined
n = int(input("Enter the number of stars: "))
square_of_stars_with_while(n)
```

```
Enter the number of stars:  5
```

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

Q 2:

Define a function that draw a square of stars on the screen.

The function name will be **sqaure_of_stars_with_for**.

The parameter will be number of stars in one side of square.

Hints:

- ask for number of stars from the user

- use for loop

```
# Q 2:

# ----- your solution here -----
def square_of_stars_with_for(n):
    for i in range(n):
        for j in range(n):
            print("*", end=" ")
        print()
# call the function you defined
n = int(input("Enter the number of stars: "))
square_of_stars_with_for(n)
```

```
Enter the number of stars:  5
```

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

Q 3:

Define a function that draw a right triangle (lower triangle) of stars on the screen.

The function name will be **lower_triangle_with_while**.

The parameter will be number of stars in one side of triangle.

Hints:

- ask for number of stars from the user
- use while loop

```
# Q 3:

# ----- your solution here -----
def lower_triangle_with_while(n):
    i = 1
    while i <= n:
        j = 1
        while j <= i:
            print("*", end=" ")
            j += 1
        print()
        i += 1
# call the function you defined
n = int(input("Masukkan angka: "))
lower_triangle_with_while(n)
```

```
Masukkan angka:  6
*
* *
* * *
* * * *
* * * * *
* * * * * *
```

Q 4:

Define a function that draw a right triangle (lower triangle) of stars on the screen.

The function name will be **lower_triangle_with_for**.

The parameter will be number of stars in one side of triangle.

Hints:

- ask for number of stars from the user
- use for loop

```
# Q 4:

# ----- your solution here -----
def lower_triangle_with_while(n):
    i = 1
    while i <= n:
        j = 1
        while j <= i:
            print("*", end=" ")
            j += 1
        print()
        i += 1
# call the function you defined
n = int(input("Masukkan angka: "))
lower_triangle_with_while(n)
```

```
Masukkan angka:  8
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * * *
```

Q 5:

Define a function that draw a right triangle (upper triangle) of stars on the screen.

The function name will be **upper_triangle_with_for**.

The parameter will be number of stars in one side of triangle.

Hints:

- ask for number of stars from the user
- use for loop

```
# Q 5:

# ----- your solution here -----
def upper_triangle_with_for(n):
    for i in range(n, 0, -1):
        for j in range(i):
            print("*", end=" ")
        print()
# call the function you defined
n = int(input("Masukkan angka: "))
upper_triangle_with_for(n)
```

```
Masukkan angka:  9
```

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

Q 6:

Define a function that draw an isosceles triangle of stars on the screen.

The function name will be **isosceles_triangle_with_for**.

The parameter will be number of stars in half of the long side of triangle.

Hints:

- ask for number of stars from the user
- call previous functions for drawing

```
# Q 6:

# ----- your solution here -----
```

```python
def lower_triangle_with_for(n):
    for i in range(1, n + 1):
        for j in range(i):
            print("*", end=" ")
        print()

def upper_triangle_with_for(n):
    for i in range(n - 1, 0, -1):
        for j in range(i):
            print("*", end=" ")
        print()

# Main function to draw the isosceles triangle
def isosceles_triangle_with_for(n):
    lower_triangle_with_for(n)
    upper_triangle_with_for(n)

# call the function you defined
n = int(input("Masukkan angka: "))
isosceles_triangle_with_for(n)
```

```
Masukkan angka:  9
```

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

Q 7:

Define a `bool` function which checks if the given number (parameter) is a prime number.

The function name will be **is_prime** and it will return True if the number is prime, False otherwise.

```python
# Q 7:

# ----- your solution here -----
def is_prime(n):
    if n < 2:
        return False

    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

# call the function you defined
n = int(input("Masukkan angka: "))
is_prime(n)
```

```
Masukkan angka:  11
```

```
True
```

Q 8:

Define a function named **prime_factors**.

The function will take an integer parameter and will find all the `prime factors` of that number.

```python
# Q 8:

# ----- your solution here -----
def prime_factors(n):
    factors = []
    while n % 2 == 0:
        factors.append(2)
        n = n // 2
    for i in range(3, int(n ** 0.5) + 1, 2):
        while n % i == 0:
            factors.append(i)
            n = n // i
    if n > 2:
        factors.append(n)

    return factors
# call the function you defined
n = int(input("Masukkan angka: "))
prime_factors(n)
```

```
Masukkan angka:  20
```

```
[2, 2, 5]
```

Q 9:

Define a function named **trees_and_fives**.

It will loop over the integers from 1 to 50, both included.

It will print the numbers but:

- If the number is a multiple of 3 it will print "Trees" instead of the number
- If the number is a multiple of 5 it will print "Fives" instead of the number
- If the number is a multiple of both 3 and 5 it will print "Trees&Fives" instead of the number

```
# Q 9:

# ----- your solution here -----
def trees_and_fives():
    # Loop through numbers from 1 to 50 (inclusive)
    for i in range(1, 51):
        if i % 3 == 0 and i % 5 == 0:
            print("Trees&Fives")
        elif i % 3 == 0:
            print("Trees")
        elif i % 5 == 0:
            print("Fives")
        else:
            print(i)
# call the function you defined
trees_and_fives()
```

```
1
2
Trees
4
Fives
Trees
7
8
Trees
Fives
11
Trees
13
14
Trees&Fives
16
17
```

```
Trees
19
Fives
Trees
22
23
Trees
Fives
26
Trees
28
29
Trees&Fives
31
32
Trees
34
Fives
Trees
37
38
Trees
Fives
41
Trees
43
44
Trees&Fives
46
47
Trees
49
Fives
```