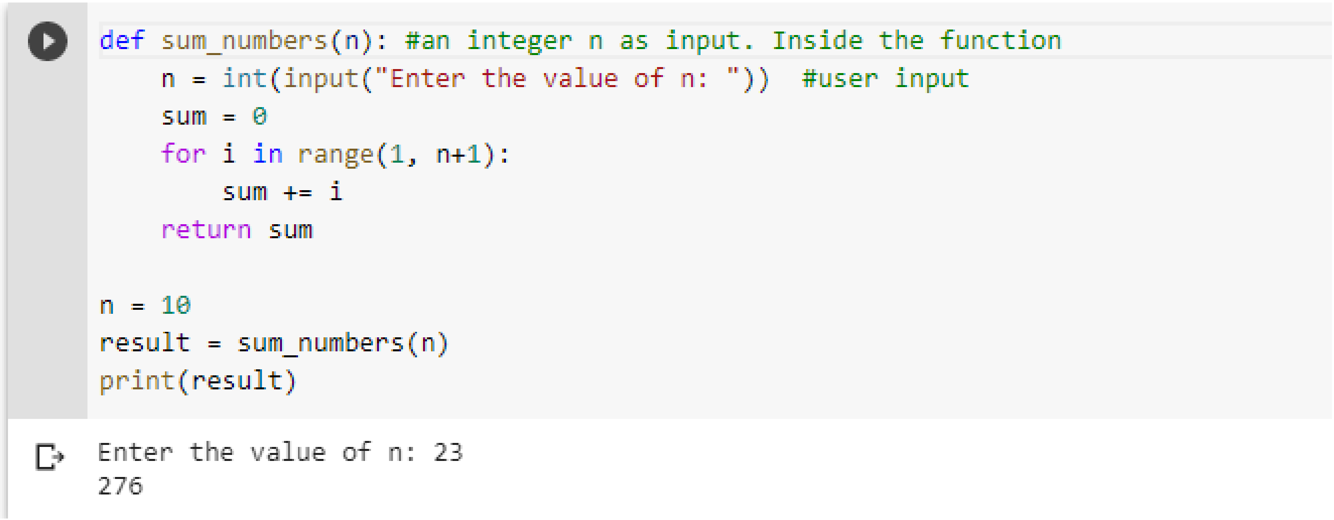
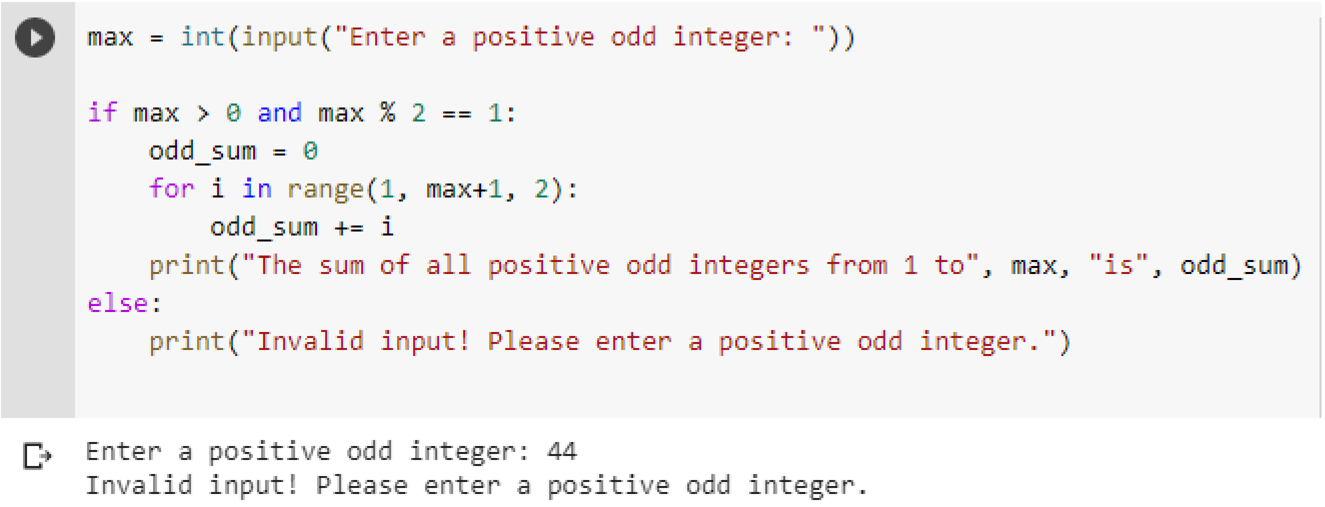
## Q1) Calculate the sum of all numbers from 1 to n.



```
def sum_numbers(n): #an integer n as input. Inside the function
    n = int(input("Enter the value of n: "))  #user input
    sum = 0
    for i in range(1, n+1):
        sum += i
    return sum

n = 10
result = sum_numbers(n)
print(result)
```

```
Enter the value of n: 23
276
```

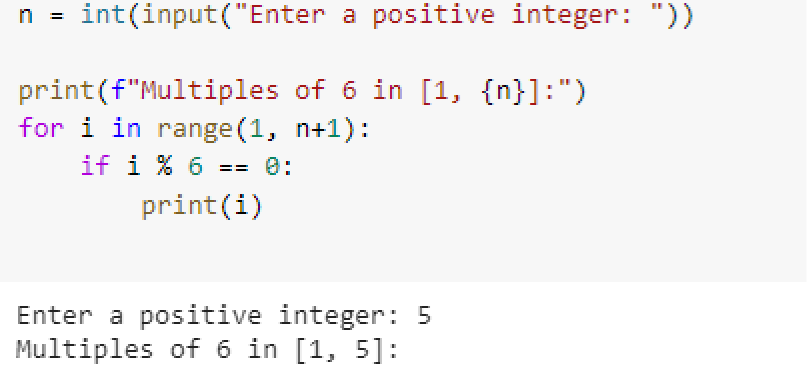
## Q2) Write a program to take an input (max) that is odd positive integer and print the sums of positive odd integers from 1 to max.



```
max = int(input("Enter a positive odd integer: "))

if max > 0 and max % 2 == 1:
    odd_sum = 0
    for i in range(1, max+1, 2):
        odd_sum += i
    print("The sum of all positive odd integers from 1 to", max, "is", odd_sum)
else:
    print("Invalid input! Please enter a positive odd integer.")
```

```
Enter a positive odd integer: 44
Invalid input! Please enter a positive odd integer.
```

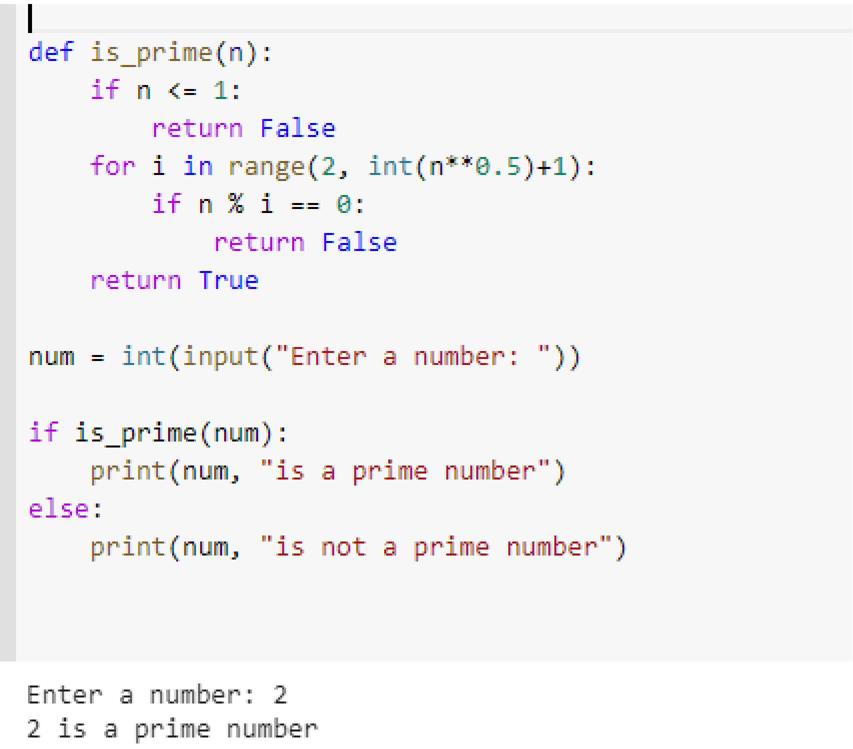
## Q3) Write a program to print all the multipliers of 6 in range[1,n], where n is positive integer.



```
n = int(input("Enter a positive integer: "))

print(f"Multiples of 6 in [1, {n}]:")
for i in range(1, n+1):
    if i % 6 == 0:
        print(i)
```

```
Enter a positive integer: 5
Multiples of 6 in [1, 5]:
```

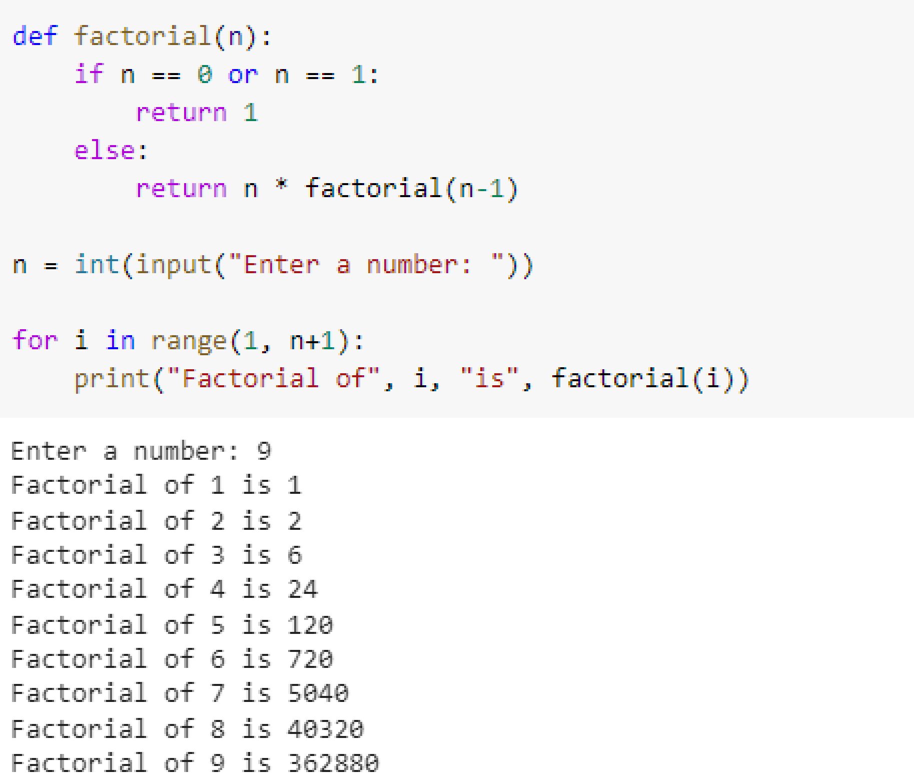
## Q4) Write a python program to check whether the given number is prime or not using function.



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

num = int(input("Enter a number: "))

if is_prime(num):
    print(num, "is a prime number")
else:
    print(num, "is not a prime number")
```

```
Enter a number: 2
2 is a prime number
```

## Q5) Write a program to display all factorial of number from 1 to n.



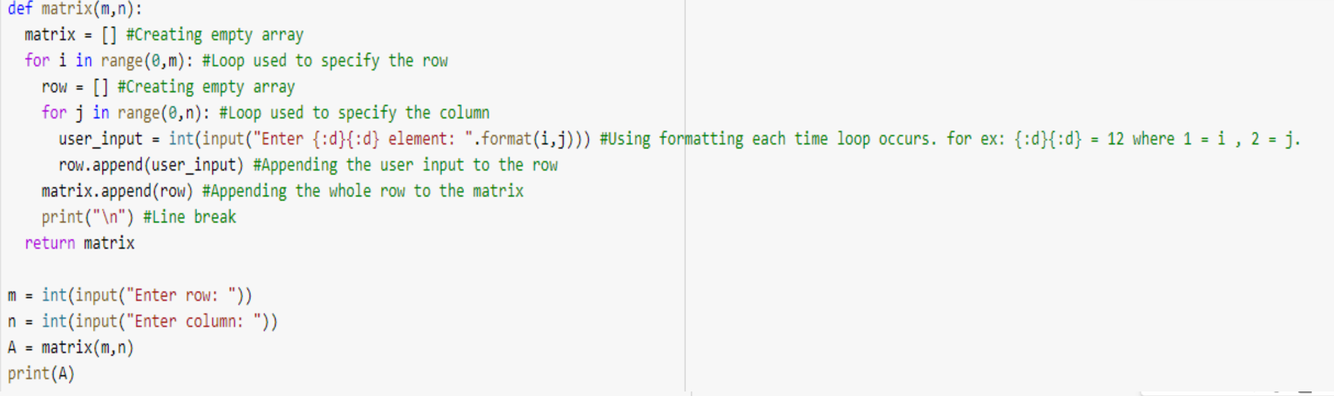
```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n-1)

n = int(input("Enter a number: "))

for i in range(1, n+1):
    print("Factorial of", i, "is", factorial(i))
```

```
Enter a number: 9
Factorial of 1 is 1
Factorial of 2 is 2
Factorial of 3 is 6
Factorial of 4 is 24
Factorial of 5 is 120
Factorial of 6 is 720
Factorial of 7 is 5040
Factorial of 8 is 40320
Factorial of 9 is 362880
```

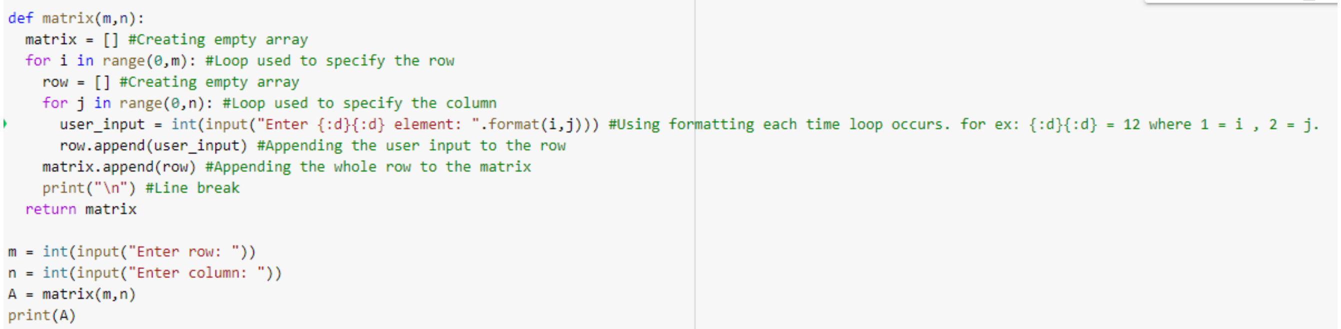
**Q6) Write a program that prints:**

**n,** $s_1(n) = \sum_{k=1}^{n} k$ **,** $s_2(n) = \sum_{k=1}^{n} k^2$ **, and**

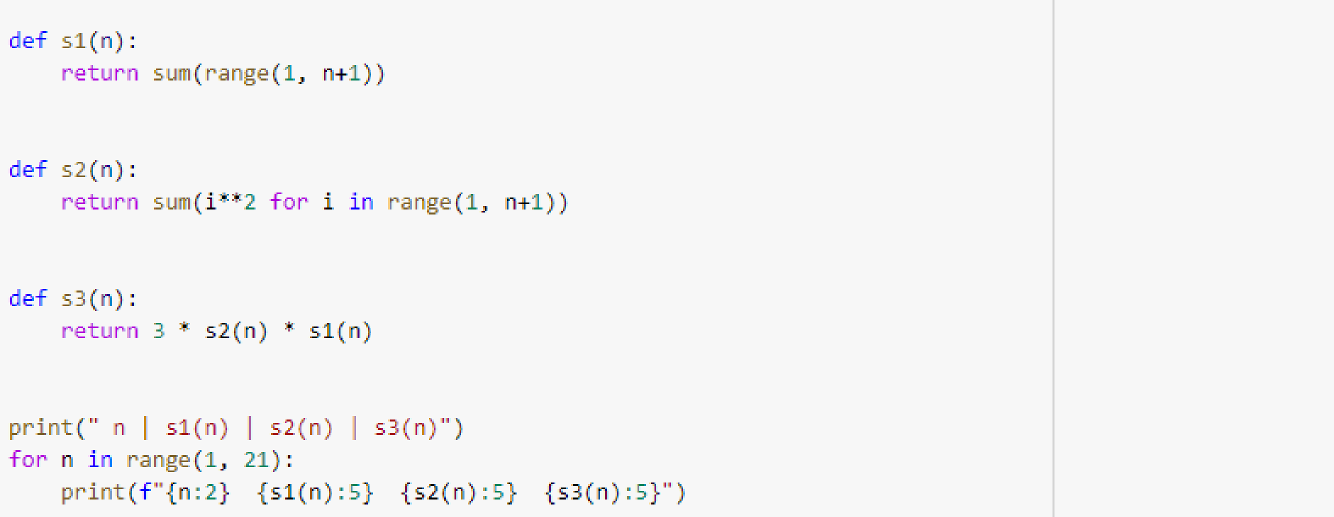
s $= \dfrac{3s_2(n)}{s_1(n)}$ **for n = 1, 2, 3, ..., 20. Try to guess the general formula for** $\sum_{k=1}^{n} k^2$ **from the resulting table.**



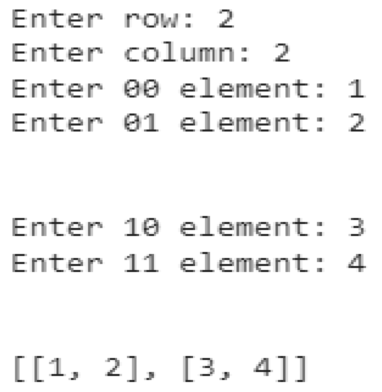
```
def matrix(m,n):
  matrix = [] #Creating empty array
  for i in range(0,m): #Loop used to specify the row
    row = [] #Creating empty array
    for j in range(0,n): #Loop used to specify the column
      user_input = int(input("Enter {:d}{:d} element: ".format(i,j))) #Using formatting each time loop occurs. for ex: {:d}{:d} = 12 where 1 = i , 2 = j.
      row.append(user_input) #Appending the user input to the row
    matrix.append(row) #Appending the whole row to the matrix
    print("\n") #Line break
  return matrix

m = int(input("Enter row: "))
n = int(input("Enter column: "))
A = matrix(m,n)
print(A)
```



```
def matrix(m,n):
  matrix = [] #Creating empty array
  for i in range(0,m): #Loop used to specify the row
    row = [] #Creating empty array
    for j in range(0,n): #Loop used to specify the column
      user_input = int(input("Enter {:d}{:d} element: ".format(i,j))) #Using formatting each time loop occurs. for ex: {:d}{:d} = 12 where 1 = i , 2 = j.
      row.append(user_input) #Appending the user input to the row
    matrix.append(row) #Appending the whole row to the matrix
    print("\n") #Line break
  return matrix

m = int(input("Enter row: "))
n = int(input("Enter column: "))
A = matrix(m,n)
print(A)
```

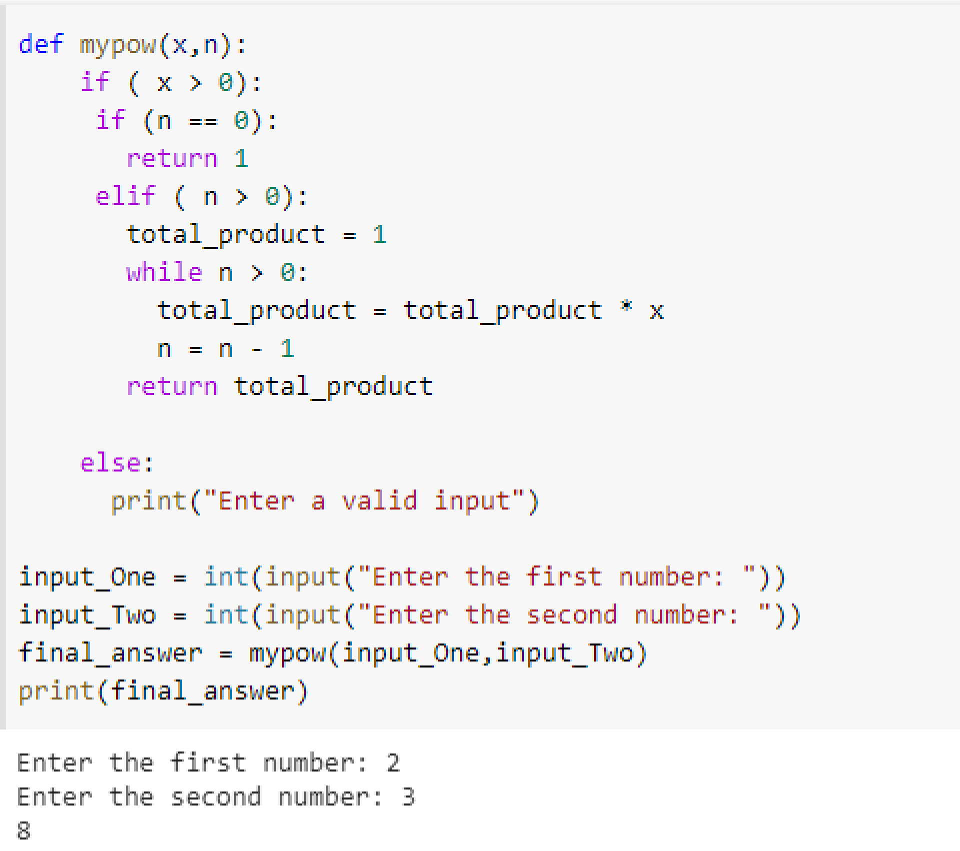


```
def s1(n):
    return sum(range(1, n+1))


def s2(n):
    return sum(i**2 for i in range(1, n+1))


def s3(n):
    return 3 * s2(n) * s1(n)


print(" n | s1(n) | s2(n) | s3(n)")
for n in range(1, 21):
    print(f"{n:2}  {s1(n):5}  {s2(n):5}  {s3(n):5}")
```



```
Enter row: 2
Enter column: 2
Enter 00 element: 1
Enter 01 element: 2


Enter 10 element: 3
Enter 11 element: 4


[[1, 2], [3, 4]]
```

**Q7) Write a function myPow(x, n) that returns $x^n$, where and n is a non-negative integer. Do not use the ** operator or the math.pow function — use one *while* loop. Hint: $x^0 = 1$.**

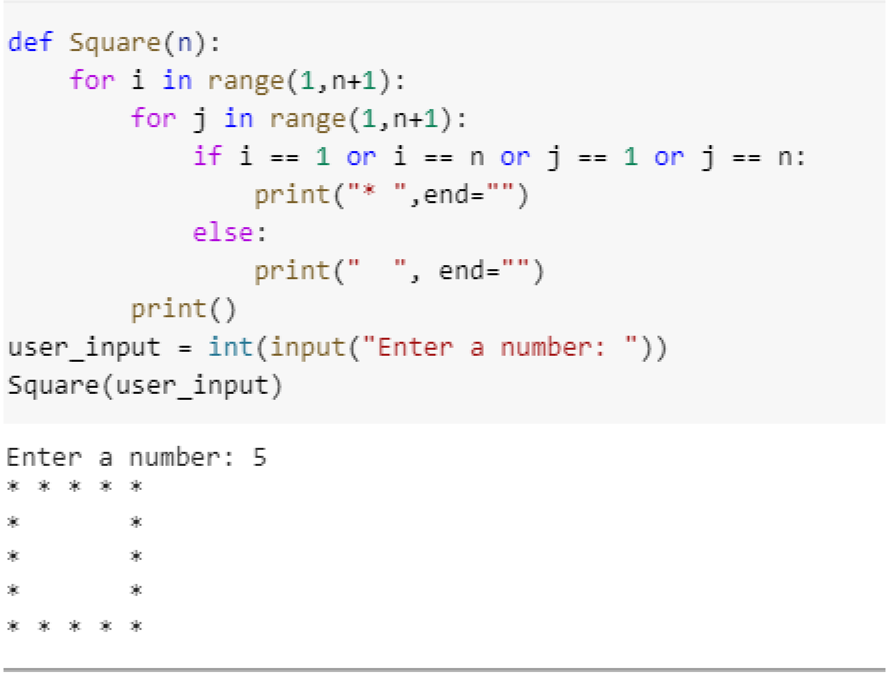


```python
def mypow(x,n):
    if ( x > 0):
     if (n == 0):
       return 1
     elif ( n > 0):
       total_product = 1
       while n > 0:
         total_product = total_product * x
         n = n - 1
       return total_product

    else:
      print("Enter a valid input")

input_One = int(input("Enter the first number: "))
input_Two = int(input("Enter the second number: "))
final_answer = mypow(input_One,input_Two)
print(final_answer)
```

```
Enter the first number: 2
Enter the second number: 3
8
```

**Q8) Write a function printSquare(n) that displays a "square" whose side has n stars. For example, for n = 5, the output should be:**

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



```
def Square(n):
    for i in range(1,n+1):
        for j in range(1,n+1):
            if i == 1 or i == n or j == 1 or j == n:
                print("* ",end="")
            else:
                print("  ", end="")
        print()
user_input = int(input("Enter a number: "))
Square(user_input)
```

```
Enter a number: 5
* * * * * 
*       * 
*       * 
*       * 
* * * * * 
```