-Presented BY Rajesh Kumar
Data Scientist

Hackathon 2.0

Major Question 1

1. Write a program to print the following pattern :

In [1]:
```
n = int(input("Enter the number of rows:"))

for n in range(1, n+1):
  for column in range(1, n+1):
    print("*",end="")
  print()

for n in range(n-1, 0, -1):
  for column in range(1, n+1):
    print("*",end="")
  print()
```

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

2. Write a program to accept 5 even and 5 odd numbers from the user and display :

- sum of even numbers
- product of odd numbers
- absolute difference of the sum and product.

Check if the final result (absolute difference) is a prime number or not.

In [4]:
```
my_list = []

#2
total = int(input("How many numbers you want to add to the list : "))

#3
for i in range(0, total):
    my_list.append(int(input("Enter : ")))

print("You have entered: ", my_list)

#4
even_sum = 0
odd_product = 1

for i in my_list:
    if(i % 2 == 0):
        even_sum += i
    else:
        odd_product *= i

#6
print("sum of all even numbers: ", even_sum)
print("Product of all odd numbers: ", odd_product)

num = abs(odd_product - even_sum)
print("Difference of the sum and product:", num)

if num > 1:
        for i in range(2, int(num/2)+1):
                if (num % i) == 0:
                        print(num, "is not a prime number")
                        break
        else:
                print(num, "is a prime number")
else:
        print(num, "is not a prime number")
```

```
How many numbers you want to add to the list : 10
Enter : 2
Enter : 4
Enter : 6
Enter : 8
Enter : 10
Enter : 1
Enter : 3
Enter : 5
Enter : 7
Enter : 9
You have entered:  [2, 4, 6, 8, 10, 1, 3, 5, 7, 9]
sum of all even numbers:  30
Product of all odd numbers:  945
Difference of the sum and product: 915
915 is not a prime number
```

3.Create a class named Item that holds data about an item in a retail store. The class should have the following three properties:

- name: the name property is a String object that holds the name of the item.
- price: the price property is a double variable that holds the item's retail price
- quantity: the quantity property is an int variable that holds the number of units currently in inventory

Write the following four methods to retrieve the values from the three fields and their current inventory value

- getName( ) returns the item name of type String
- getPrice( ) returns the price of the item of type double
- getQuantity( ) returns the number of quantities of type int
- getValue( ) that returns the current inventory value (quantity * price) of type double

In [5]:
```
class item:
  def __init__(self, name:str, price:float, quantity:int):
    self.name = name
    self.price = price
    self.quantity = quantity
  def getname(self):
    return self.name
  def getprice(self):
    return self.price
  def getquantity(self):
    return self.quantity
  def getvalue(self):
    self.value= self.price* self.quantity
    return self.value

i = item("Jeans", 350, 3)
print("Name of item: ", i.getname())
print("Price of item: ", i.getprice(), "Rs.")
print("Quantity of items: ", i.getquantity(), "Qty")
print("Current inventory value: ", i.getvalue(), "Rs.")
```

```
Name of item:  Jeans
Price of item:  350 Rs.
Quantity of items:  3 Qty
Current inventory value:  1050 Rs.
```

Major Question 2

1. Ask the user number of rows to be generated of a series. Suppose user enters no. of rows = 5 then the series shall be :

In [6]:
```
num_rows = int(input("Enter the number of rows: "))
for i in range(1, num_rows+1):
 print(int("9" * i))
```

```
Enter the number of rows: 5
9
99
999
9999
99999
```

2. Write a program to accept a number from the user and check whether the number entered is prime or not.

In [8]:
```
num = int(input("Enter The Number: "))
# If given number is greater than 1
if num > 1:
        # Iterate from 2 to n / 2
        for i in range(2, int(num/2)+1):
                # If num is divisible by any number between
                # 2 and n / 2, it is not prime
                if (num % i) == 0:
                        print(num, "is not a prime number")
                        break
        else:
                print(num, "is a prime number")
else:
        print(num, "is not a prime number")
```

```
Enter The Number: 7
7 is a prime number
```

OR Another way

In [11]:
```
import math

n=int(input("Enter The Number: "))

def is_prime(n):
        if n < 2:
                return (n, "is not a prime number")
        i = 2
        while i*i <= n:
                if n % i == 0:
                        return (n, "is not a prime number")
                i += 1
        return (n, "is a prime number")

print(is_prime(n))
```

```
Enter The Number: 7
(7, 'is a prime number')
```

3. Continued from Major Question 1. Write a separate class called Inventory with methods

- generate() - creates three Item objects
- getDetails() - produces a neatly formatted table of the store's inventory displaying

the three items, their current inventory value, and the total inventory value for the store.

In [12]:
```
# Create a class named Item
class Item:
  # Create a constructor
  def __init__(self, name:str, price:float, quantity:int):
    self.name = name
    self.price = price
    self.quantity = quantity

    # Create a four Methods
  def getname(self):
    return self.name
  def getprice(self):
    return self.price
  def getquantity(self):
    return self.quantity
  def getvalue(self):
    self.value= self.price* self.quantity
    return self.value

# Create a separate class named Inventory
class Inventory(Item):
  def generate():
    inven = []
    inven.append(Inventory("Stapler  |", 2.25, 15))
    inven.append(Inventory("Paper   |", 32.99, 255))
    inven.append(Inventory("Binder  |", 4.75, 9))
    return inven


  def getDetails():
        items = Inventory.generate()
        total_value = 0
        print("{:<10} {:<9.5} {:<10} {:<10}".format("Name    |", "Price", "Quantity", "Value"))
        print("========|=========|==========|==========")
        for i in items:
            print("{:<10} {:<9.2f} {:<10} {:<9.2f}".format(i.getname(), i.getprice(), i.getquantity(), i.getvalue()))
            print("--------|---------|----------|----------")
            total_value += i.getvalue()
        print("Total Inventory Value: {:.2f}".format(total_value))

Inventory.getDetails()
```

```
Name     |  Price     Quantity   Value
========|=========|==========|==========
Stapler  |  2.25      15         33.75
--------|---------|----------|----------
Paper    |  32.99     255        8412.45
--------|---------|----------|----------
Binder   |  4.75      9          42.75
--------|---------|----------|----------
Total Inventory Value: 8488.95
```

Print a Table of the question above in Proper way.

In [13]:
```
# import the module
import pandas as pd


# create a DataFrame
df = pd.DataFrame({"Name":['Stapler', 'Paper', 'Binder', "Total Inventory Value"],
                   "Price":[2.25, 32.99, 4.75, "-"],
                   "Quantity":[15, 225, 9, "-"],
                   "Value":[33.75, 8412.45, 42.75, 8488.95]})

# making a green border
df.style.set_table_styles([{'selector' : ' ',
                                                'props' : [('border',
                                                            '5px solid green')]}])
```

Out[13]:

	Name	Price	Quantity	Value
0	Stapler	2.250000	15	33.750000
1	Paper	32.990000	225	8412.450000
2	Binder	4.750000	9	42.750000
3	Total Inventory Value	-	-	8488.950000