UNIT-2

Write a program to determine a given number is ‘odd’ or ‘even’ and print the following message “Number is ODD” or “Number is Even”.

In [1]:

```
number = int(input("Enter a number: "))
if number % 2 == 0:
    print("Number is Even")
else:
    print("Number is Odd")
```

```
Enter a number: 5
Number is Odd
```

Write a program to check if the input number is positive, negative or zero.

In [2]:

```
number = float(input("Enter a number: "))
if number > 0:
    print("Number is Positive")
elif number < 0:
    print("Number is Negative")
else:
    print("Number is Zero")
```

```
Enter a number: 5
Number is Positive
```

Write a program to find the maximum number among the three input numbers.

In [3]:

```
# Taking three numbers as input
a = float(input("Enter first number: "))
b = float(input("Enter second number: "))
c = float(input("Enter third number: "))
if a>b and a>c:
    print(f"max number is {a}")
elif b>c:
    print(f"max number is {b}")
else:
    print(f"max number is {c}")
```

```
Enter first number: 5
Enter second number: 7
Enter third number: 3
max number is 7.0
```

Write a program to check if year is a leap year or not (Nested If).

In [4]:

```
year=int(input("enter year: "))
if year%4==0:
    if year%400==0 and year%100!=0:
        print(f"{year} is leap year")
    else:
        print(f"{year} is not leap year")
else:
    print(f"{year} is not leap year")
```

```
enter year: 1800
1800 is not leap year
```

Write a program to find sum of first N natural numbers given by user.

In [6]:

```
n= int(input("enter number: "))
sum=0
for i in range(1,n+1):
    sum+=i
print(f"sum of n number {sum}")
```

```
enter number: 10
sum of n number 55
```

Write a program to find average of first N natural numbers given by user.

In [8]:

```
n= int(input("enter number: "))
sum=0
count=0
for i in range(1,n+1):
    sum+=i
    count+=1
print(f"sum of n number {sum}")
print(f"Avg of n number {sum/count}")
```

```
enter number: 10
sum of n number 55
Avg of n number 5.5
```

Write a python program to read three numbers (a,b,c) and check how many numbers between ‘a’ and ‘b’ are divisible by ‘c’

In [12]:

```
a = int(input("Enter the first number (a): "))
b = int(input("Enter the second number (b): "))
c = int(input("Enter the number to check divisibility (c): "))

count = 0
```

```
for i in range(a + 1, b):
  if i % c == 0:
    count += 1
    print(f"number is divisible by {c}:{i}" )

print(f"Total number is :{count}")
```

```
Enter the first number (a): 1
Enter the second number (b): 20
Enter the number to check divisibility (c): 3
number is divisible by 3:3
number is divisible by 3:6
number is divisible by 3:9
number is divisible by 3:12
number is divisible by 3:15
number is divisible by 3:18
Total number is :6
```

Write a Python program that prints all the numbers from 0 to 6 except 3 and 6.

In [13]:

```
for i in range(7):
    if i != 3 and i != 6:
        print(i)
```

```
0
1
2
4
5
```

Write a Python program to print the multiplication table of given number by user.

In [14]:

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

for i in range(1, 11):
    print(num, "x", i, "=", num * i)
```

```
Enter a number: 15
15 x 1 = 15
15 x 2 = 30
15 x 3 = 45
15 x 4 = 60
15 x 5 = 75
15 x 6 = 90
15 x 7 = 105
15 x 8 = 120
15 x 9 = 135
15 x 10 = 150
```

Write a program to find the factorial of a number provided by the user.

In [15]:

```
num = int(input("Enter a number: "))
factorial = 1
for i in range(1, num + 1):
    factorial *= i
print("Factorial of", num, "is:", factorial)
```

```
Enter a number: 10
Factorial of 10 is: 3628800
```

Write a python program to display the Fibonacci sequence up to n-th term.

In [16]:

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

a, b = 0, 1
count = 0

if n <= 0:
    print("Please enter a positive integer")
elif n == 1:
    print("Fibonacci sequence upto", n, ":", a)
else:
    print("Fibonacci sequence:")
    while count < n:
        print(a, end=" ")
        nth = a + b
        a = b
        b = nth
        count += 1
```

```
Enter the number of terms: 10
Fibonacci sequence:
0 1 1 2 3 5 8 13 21 34
```

Write a program to take 10 values from keyboard using loop and print their average on the screen

In [17]:

```
total = 0
for i in range(10):
    num = float(input(f"Enter number {i+1}: "))
    total += num
average = total / 10
print("Average of 10 numbers is:", average)
```

```
Enter number 1: 1
Enter number 2: 2
Enter number 3: 3
Enter number 4: 4
Enter number 5: 5
Enter number 6: 6
Enter number 7: 7
Enter number 8: 8
Enter number 9: 9
Enter number 10: 10
Average of 10 numbers is: 5.5
```

Write a program to reverse a number.

In [18]:

```
num = int(input("Enter a number: "))
reverse_num = 0
```

```
while num > 0:
    digit = num % 10
    reverse_num = (reverse_num * 10) + digit
    num //= 10
print("Reversed number is:", reverse_num)
```

```
Enter a number: 1265
Reversed number is: 5621
```

Write a program to check whether a number is Armstrong number or not.

In [19]:

```
num = int(input("Enter a number: "))
order = len(str(num))
temp = num
sum_armstrong = 0

while temp > 0:
    digit = temp % 10
    sum_armstrong += digit ** order
    temp //= 10

if num == sum_armstrong:
    print(num, "is an Armstrong number")
else:
    print(num, "is not an Armstrong number")
```

```
Enter a number: 153
153 is an Armstrong number
```

Write a program to check if a number is prime or not.

In [21]:

```
num = int(input("Enter a number: "))
is_prime = True

if num <= 1:
    is_prime = False
else:
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            is_prime = False
            break

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

```
Enter a number: 6
6 is not a prime number
```

Write a program to print prime numbers between given interval from user

In [22]:

```
start = int(input("Enter the start of the interval: "))
end = int(input("Enter the end of the interval: "))
```

```
print("Prime numbers between", start, "and", end, "are:")
for num in range(start, end + 1):
    if num > 1:
        is_prime = True
        for i in range(2, int(num ** 0.5) + 1):
            if num % i == 0:
                is_prime = False
                break
        if is_prime:
            print(num)
```

```
Enter the start of the interval: 1
Enter the end of the interval: 100
Prime numbers between 1 and 100 are:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

Draw a pattern using a python program:

*

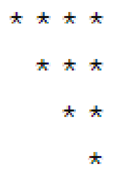
*

In [25]:

```
# Number of rows for the pattern
rows = int(input("enter number of rows: "))

# Nested loop to create the pattern
for i in range(1, rows + 1):
    for j in range(1, i + 1):
        print("*", end=" ")
    print()
```

```
enter number of rows: 5
*
* *
* * *
* * * *
* * * * *
```

Draw a pattern:

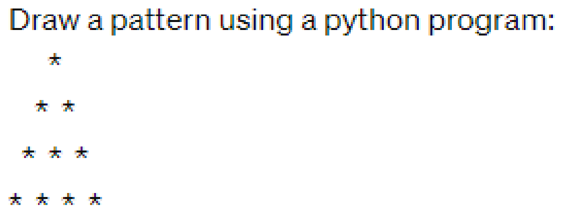


```
* * * *
  * * *
    * *
      *
```

In [26]:

```python
# Number of rows for the pattern
rows = int(input("enter number of rows: "))

# Nested loop to create the pattern
for i in range(rows):
    for j in range(i):
        print(" ", end=" ")
    for k in range(rows - i):
        print("*", end=" ")
    print()
```

```
enter number of rows: 5
* * * * *
  * * * *
    * * *
      * *
        *
```



Draw a pattern using a python program:

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

In [35]:

```python
# Number of rows for the pattern
rows = int(input("enter number of rows: "))

# Nested loop to create the pattern
for i in range(rows):
    # Printing spaces before the first set of asterisks in each row
    for j in range(rows - i - 1):
        print(" ", end="")

    # Printing asterisks in a pattern for each row
    for k in range(i + 1):
        print(" *", end="")

    print()
```

```
enter number of rows: 5
     * 
    * * 
   * * * 
  * * * * 
 * * * * * 
```

Draw a pattern:

* * * *

* * *

* *

*

In [37]:

```python
# Number of rows for the pattern
rows = int(input("enter number of rows: "))

# Nested loop to create the pattern
for i in range(rows, 0, -1):
    for j in range(i):
        print("*", end=" ")
    print()
```

```
enter number of rows: 5
* * * * * 
* * * * 
* * * 
* * 
* 
```

"Draw a pattern using a python program:

1 2 3 4 5

1 2 3 4

1 2 3

1 2

1"

In [38]:

```python
# Number of rows for the pattern
rows = int(input("enter number of rows: "))

# Nested loop to create the pattern
for i in range(rows, 0, -1):
    for j in range(1, i + 1):
        print(j, end=" ")
    print()
```

```
enter number of rows: 5
1 2 3 4 5 
1 2 3 4 
1 2 3 
1 2 
1 
```

Draw a pattern using a python program:

1

1 2

1 2 3

1 2 3 4

In [39]:

```
# Number of rows for the pattern
rows = int(input("enter number of rows: "))

# Nested loop to create the pattern
for i in range(1, rows + 1):
    for j in range(1, i + 1):
        print(j, end=" ")
    print()
```

```
enter number of rows: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Draw a pattern using a python program:

1

2 2

3 3 3

4 4 4 4

In [40]:

```
# Number of rows for the pattern
rows = int(input("enter number of rows: "))

# Nested loop to create the pattern
for i in range(1, rows + 1):
    for j in range(i):
        print(i, end=" ")
    print()
```

```
enter number of rows: 4
1
2 2
3 3 3
4 4 4 4
```

"Draw a pattern using a python program:

*

# #

* * *

#"

In [41]:

```
# Number of rows for the pattern
rows = int(input("enter number of rows: "))

# Nested loop to create the pattern
for i in range(1, rows + 1):
    for j in range(i):
        if i % 2 == 0:
            print("#", end=" ")
        else:
            print("*", end=" ")
    print()
```

```
enter number of rows: 5
*
# #
* * *
# # # #
* * * * *
```

"Draw a pattern using a python program:

1

0 1

1 0 1

0 1 0 1"

In [42]:

```
# Number of rows for the pattern
rows = int(input("enter number of rows: "))

# Nested loop to create the pattern
for i in range(rows):
    for j in range(i + 1):
        if (i + j) % 2 == 0:
            print("1", end=" ")
        else:
            print("0", end=" ")
    print()
```

```
enter number of rows: 5
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

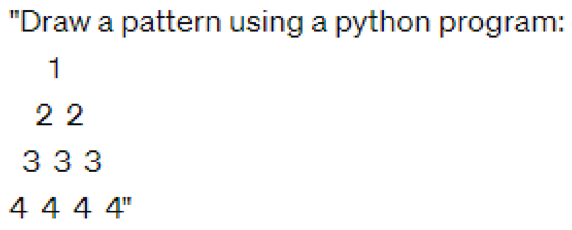
Draw a pattern using a python program:

```
   1
  1 2
 1 2 3
1 2 3 4
```

In [47]:

```
# Number of rows for the pattern
rows = int(input("enter number of rows: "))
# Nested loop to create the pattern
for i in range(1, rows + 1):
    # Print spaces at the beginning of each row
    for j in range(rows - i):
        print(" ", end=" ")

    # Print numbers in ascending order
    for k in range(1, i + 1):
        print(k," ", end=" ")

    print()
```

```
enter number of rows: 5
        1   
      1   2   
    1   2   3   
  1   2   3   4   
1   2   3   4   5   
```



"Draw a pattern using a python program:

1

2 2

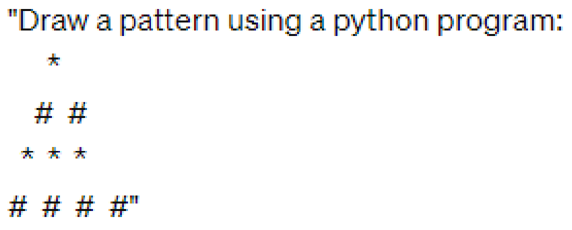
3 3 3

4 4 4 4"

In [51]:

```python
# Number of rows for the pattern
rows = int(input("enter number of rows: "))

# Nested loop to create the pattern
for i in range(1, rows + 1):
    # Print spaces at the beginning of each row
    for j in range(rows - i):
        print(" ", end=" ")

    # Print numbers in the pattern
    for k in range(i):
        print(i," ", end=" ")

    print()
```

```
enter number of rows: 5
        1   
      2   2   
    3   3   3   
  4   4   4   4   
5   5   5   5   5   
```



"Draw a pattern using a python program:

*

# #

* * *

#"

In [52]:

```python
# Number of rows for the pattern
rows = int(input("enter number of rows: "))

# Nested loop to create the pattern
for i in range(1, rows + 1):
    # Print spaces at the beginning of each row
    for j in range(rows - i):
        print(" ", end=" ")

    # Print characters in the pattern
    for k in range(i):
        if i % 2 == 0:
            print("#"," ", end=" ")
        else:
            print("*"," ", end=" ")

    print()
```

```
enter number of rows: 5
        *
      #   #
    *   *   *
  #   #   #   #
*   *   *   *   *
```

"Gross Pay, Annual Income and Income Tax Calculator Write a Python Program to make the gross pay, annual income and income tax calculator using following data.

The gross pay consists of Basic Pay, House Rent Allowance (hra), Dearness Allowance (dra), other allowances and professional tax and provident fund. Gross Pay= Basic Pay+ House Rent Allowance (hra) + Dearness Allowance (dra) +other allowances +Transport Allowance (TA)– Professional tax –Employees' Provident fund (EPF) Basic Pay for different grade levels are indicated in table given. The Professional tax remains constant and that is equal to 200 Rs. for each grade levels and each month. House Rent Allowance (hra) varies as per the city- For Class 1 Cities it is 0.3 times of Basic Pay of each grade levels, for Class 2 Cities it is 0.2 times of Basic Pay of each grade levels, for Class 3 Cities it is 0.1 times of Basic Pay of each grade levels, Dearness Allowance (dra)= 0.5 times of Basic Pay of each grade levels, Other allowances are given in table which varies according to different grade levels, Provident Fund= 0.11 times of Basic Pay for each grade levels, Transport Allowance remains constant as 900 Rs. for each levels. For different grade pays:"

"The gross pay calculated is only for one month. After calculating Gross Pay of each employee calculate the annual pay for employee by multiplying gross pay calculated, by 12. So, Annual Pay of an employee=Gross Pay of an employee*12 From Annual Pay of an Employee Calculate the income tax as per the slabs of India Income Tax 2022-23 given below. Tax Slabs for AY 2022-23"

"Input & Output: Enter the grade_level (A,B,C,D,E or F:)A city 1 is a tier 1 (metro), city 2 is tier 2 and city 3 is tier 3 Enter the city (1,2 or 3)1 Gross Pay of an Employee is: 110100.0 Annual income of an employee is: 1321200.0 Income Tax to be paid by an employee is: 142800.0 "

Gross Pay, Annual Income and Income Tax Calculator

Write a Python Program to make the gross pay, annual income and income tax calculator using following data.

The gross pay consists of Basic Pay, House Rent Allowance (hra), Dearness Allowance (dra), other allowances and professional tax and provident fund.

Gross Pay= Basic Pay+ House Rent Allowance (hra) + Dearness Allowance (dra) +other allowances +Transport Allowance (TA)– Professional tax –Employees' Provident fund (EPF)

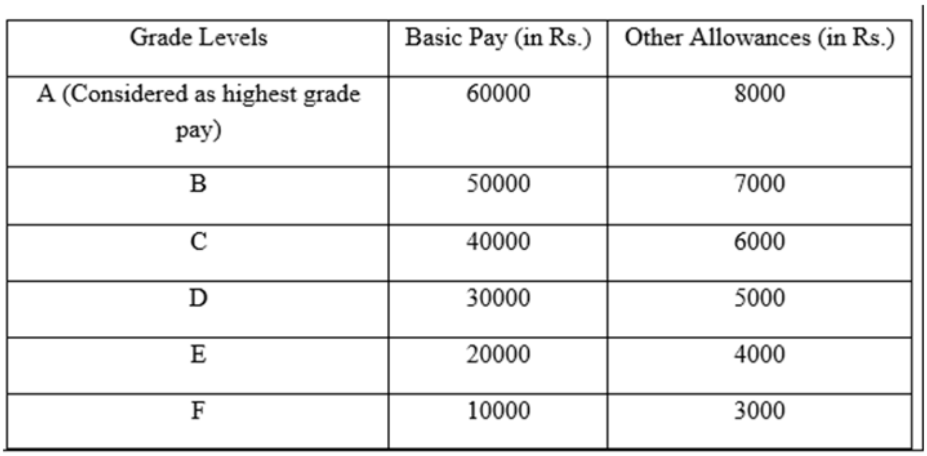
Basic Pay for different grade levels are indicated in table given.

The Professional tax remains constant and that is equal to 200 Rs. for each grade levels and each month.

House Rent Allowance (hra) varies as per the city- For Class 1 Cities it is 0.3 times of Basic Pay of each grade levels, for Class 2 Cities it is 0.2 times of Basic Pay of each grade levels, for Class 3 Cities it is 0.1 times of Basic Pay of each grade levels,

Dearness Allowance (dra)= 0.5 times of Basic Pay of each grade levels, Other allowances are given in table which varies according to different grade levels, Provident Fund= 0.11 times of Basic Pay for each grade levels, Transport Allowance remains constant as 900 Rs. for each levels.

For different grade pays:



Grade Levels	Basic Pay (in Rs.)	Other Allowances (in Rs.)
A (Considered as highest grade pay)	60000	8000
B	50000	7000
C	40000	6000
D	30000	5000
E	20000	4000
F	10000	3000

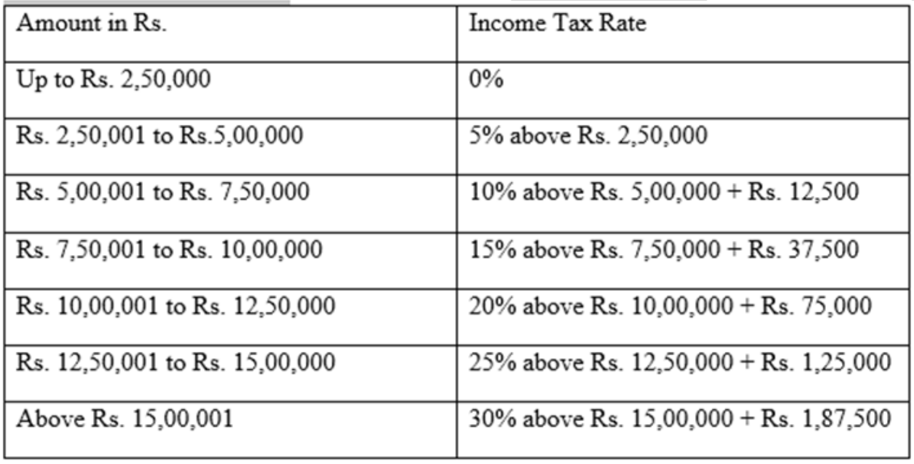
The gross pay calculated is only for one month.

After calculating Gross Pay of each employee calculate the annual pay for employee by multiplying gross pay calculated, by 12.

So, Annual Pay of an employee=Gross Pay of an employee*12

From Annual Pay of an Employee Calculate the income tax as per the slabs of India Income Tax 2022-23 given below.

Tax Slabs for AY 2022-23



Amount in Rs.	Income Tax Rate
Up to Rs. 2,50,000	0%
Rs. 2,50,001 to Rs.5,00,000	5% above Rs. 2,50,000
Rs. 5,00,001 to Rs. 7,50,000	10% above Rs. 5,00,000 + Rs. 12,500
Rs. 7,50,001 to Rs. 10,00,000	15% above Rs. 7,50,000 + Rs. 37,500
Rs. 10,00,001 to Rs. 12,50,000	20% above Rs. 10,00,000 + Rs. 75,000
Rs. 12,50,001 to Rs. 15,00,000	25% above Rs. 12,50,000 + Rs. 1,25,000
Above Rs. 15,00,001	30% above Rs. 15,00,000 + Rs. 1,87,500

Input & Output:

Enter the grade_level (A,B,C,D,E or F:)A

city 1 is a tier 1 (metro), city 2 is tier 2 and city 3 is tier 3

Enter the city (1,2 or 3)1

Gross Pay of an Employee is: 110100.0

Annual income of an employee is: 1321200.0

Income Tax to be paid by an employee is: 142800.0

In [58]:

```
# Constants for basic pay and other allowances for each grade level
basic_pay_A = 60000
basic_pay_B = 50000
```

```
basic_pay_C = 40000
basic_pay_D = 30000
basic_pay_E = 20000
basic_pay_F = 10000

other_allowances_A = 8000
other_allowances_B = 7000
other_allowances_C = 6000
other_allowances_D = 5000
other_allowances_E = 4000
other_allowances_F = 3000

# Constants for tax calculation
Professional_tax = 200  # Professional tax
Transport_allowance = 900  # Transport allowance
EPF_rate = 0.11  # Employees’ Provident Fund rate

# Get user inputs for grade level and city
grade_level = input("Enter the grade_level (A, B, C, D, E, or F): ")
city = int(input("Enter the city (1, 2, or 3): "))

# Calculate HRA, DRA, Gross Pay, and other values based on inputs
if grade_level == 'A':
    basic = basic_pay_A
    other_allowances = other_allowances_A
elif grade_level == 'B':
    basic = basic_pay_B
    other_allowances = other_allowances_B
elif grade_level == 'C':
    basic = basic_pay_C
    other_allowances = other_allowances_C
elif grade_level == 'D':
    basic = basic_pay_D
    other_allowances = other_allowances_D
elif grade_level == 'E':
    basic = basic_pay_E
    other_allowances = other_allowances_E
elif grade_level == 'F':
    basic = basic_pay_F
    other_allowances = other_allowances_F

if city == 1:
    HRA = 0.3 * basic
elif city == 2:
    HRA = 0.2 * basic
elif city == 3:
    HRA = 0.1 * basic

DRA = 0.5 * basic
EPF = EPF_rate * basic

# Calculate Gross Pay
Gross_pay = (basic + HRA + DRA + other_allowances +
             Transport_allowance - Professional_tax - EPF)

# Calculate Annual Income
Annual_income = Gross_pay * 12

# Calculate Income Tax based on slabs
if Annual_income <= 250000:
    Income_tax = 0
elif 250000 < Annual_income <= 500000:
    Income_tax = (Annual_income - 250000) * 0.05
elif 500000 < Annual_income <= 750000:
    Income_tax = 12500 + (Annual_income - 500000) * 0.1
elif 750000 < Annual_income <= 1000000:
    Income_tax = 37500 + (Annual_income - 750000) * 0.15
elif 1000000 < Annual_income <= 125000:
    Income_tax = 75000 + (Annual_income - 1000000) * 0.2
elif 1250000 < Annual_income <= 1500000:
    Income_tax = 125000 + (Annual_income - 1250000) * 0.25
else:
    Income_tax = 187500 + (Annual_income - 1250000) * 0.3

# Display results
print(f"Gross Pay of an Employee is: {Gross_pay}")
print(f"Annual income of an employee is: {Annual_income}")
print(f"Income Tax to be paid by an employee is: {Income_tax}")
```

```
Enter the grade_level (A, B, C, D, E, or F): A
Enter the city (1, 2, or 3): 1
Gross Pay of an Employee is: 110100.0
Annual income of an employee is: 1321200.0
Income Tax to be paid by an employee is: 142800.0
```

"Write a python program to print all numbers between 1 and 100 (including 1 and 100) that are both, Disarium and Harshad numbers.

- A number is said to be a Disarium number when the sum of its digit raised to the power of their respective positions becomes equal to the number itself.
- For example, 175 is a Disarium number as follows:
- 11+ 72 + 53 = 1+ 49 + 125 = 175

- A harshad number is a number that is divisible by the sum of its digits. E.g., the number 18 is a harshad number, because the sum of the digits 1 and 8 is 9 (1 + 8 = 9), and 18 is divisible by 9.

In [106…

```
for number_to_check in range(1,100):   # Replace this number with the one you want to

    num = number_to_check
    digit_count = 0
    temp = number_to_check

    # Count the number of digits in the given number
    while temp:
        temp //= 10
        digit_count += 1

    temp = number_to_check
    sum_disarium = 0

    # Calculate the sum of digits raised to their respective positions
    while temp>0:
        digit = temp % 10
        sum_disarium += digit ** digit_count
        temp //= 10
        digit_count -= 1

    # Check if the sum is equal to the original number
    #if sum_disarium == number_to_check:
        #print(f"{number_to_check} is a Disarium number")
    digit_sum = 0
    temp = number_to_check
    while temp > 0:
        digit_sum += temp % 10
        temp //= 10

    # Check if the number is divisible by the sum of its digits
    #if number_to_check % digit_sum == 0:
        #print(f"{number_to_check} is a Harshad number")
    # check both harshad and Disarium number
    if number_to_check % digit_sum == 0 and sum_disarium == number_to_check:
        print(f"{number_to_check} is a Harshad and disarium number")
```

```
1 is a Harshad and disarium number
2 is a Harshad and disarium number
3 is a Harshad and disarium number
4 is a Harshad and disarium number
5 is a Harshad and disarium number
6 is a Harshad and disarium number
7 is a Harshad and disarium number
8 is a Harshad and disarium number
9 is a Harshad and disarium number
```

"Ask the user to enter 10 test scores. Write a program to do the following:

- a) If user enters score greater than 100, then give warning to user that entered score is more than 100 and take that input again from user.
- b) Print out the highest and lowest scores.
- c) Print out the average of the scores.
- d) Print out the second largest score.
- e) Drop the two lowest scores and print out the average of the rest of them. Note: Use of Python Data structures like string, list, tuple etc. and their inbuilt function is not allowed.

For Ex.

If Input is like following:

Enter Test Score: 80

Enter Test Score: 65

Enter Test Score: 98

Enter Test Score: 70

Enter Test Score: 93

Enter Test Score: 130

Entered score is more than hundred, so enter again

Enter Test Score: 95

Enter Test Score: 50

Enter Test Score: 40

Enter Test Score: 75

Enter Test Score: 72

Output should be:

Highest Score is: 98

Lowest Score is: 40

Average Test Score is: 73.8

Second Largest Score is: 95

Average after dropping the two lowest scores: 81.0

In [72]:

```
sum=0
count=1
f_max=0
f_min=100
while count<11:
    n=int(input(f"enter {count} score"))
    if n>100:
        print("enter valid Score")
        continue
    count+=1
    sum+=n
    if n>f_max:
        s_max=f_max
        f_max=n
    elif n>s_max and s_max!=f_max:
        s_max=n
    if n<f_min:
        s_min=f_min
        f_min=n
    elif n<s_min and s_min!=f_min:
        s_min=n
print("highest score",f_max)
print("lowest score",f_min)
print("Avg test score",sum/count)
print("Second Largest Score",s_max)
print("Average after dropping the two lowest scores",(sum-f_min-s_min)/(count-2))
```

```
enter 1 score80
enter 2 score70
enter 3 score60
enter 4 score150
enter valid Score
enter 4 score50
enter 5 score70
enter 6 score20
enter 7 score89
enter 8 score90
enter 9 score45
enter 10 score67
highest score 90
lowest score 20
Avg test score 58.27272727272727
Second Largest Score 89
Average after dropping the two lowest scores 64.0
```

"Write a program to encode a number by changing the digits in the given positive integer by user. The rule for changing the digits in number will be:

If the digit in number is between 0 to 8 than replace the number with 1 to 9 respectively. (incrementing each digit by +1).

If the digit is 9, then replace it with 0.

To encode a number, replace digits in following manner:

For example:

Input: 31590218

Output: The number after encoding is: 42601329

For example:

Input: 9259

Output: The number after encoding is: 360

For example:

Input: 65217001

Output: The number after encoding is: 76328112

Original Digit in Number	New Digit after Encoding
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	0

Note: Use of Python Data structures like string, list, tuple etc. and their inbuilt function is not allowed."

In [73]:

```
# Get the positive integer from the user
number = int(input("Enter a positive integer: "))

# Initialize variables
encoded_number = 0
multiplier = 1

# Encode the number based on the given rules
while number > 0:
    remainder = number % 10
    if remainder == 9:
        encoded_digit = 0
    else:
        encoded_digit = remainder + 1
    encoded_number += encoded_digit * multiplier
    multiplier *= 10
    number //= 10

# Display the encoded number
print(f"The number after encoding is: {encoded_number}")
```

```
Enter a positive integer: 1234567890
The number after encoding is: 2345678901
```

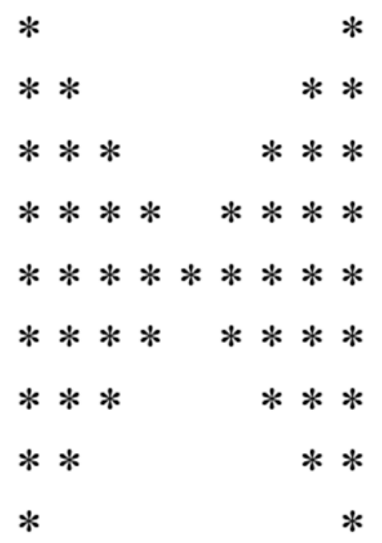
"Write a python program to swap first and last digits of a number using loop. (for example: input = 123456 then output=623451)"

In [77]:

```
# Get the number from the user
number = int(input("Enter a number: "))

# Find the number of digits in the number
temp = number
digit_count = 0
while temp > 0:
    temp //= 10
    digit_count += 1

# Extract the first and last digits
last_digit = number % 10
first_digit = number // (10 ** (digit_count - 1))

# Calculate the remaining part of the number after excluding the first and last digits
middle_part = (number % (10 ** (digit_count - 1))) // 10
# Construct the number after swapping the first and last digits
new_number = last_digit * (10 ** (digit_count - 1)) + middle_part * 10 + first_digit

# Display the number after swapping the first and last digits
print(f"The number after swapping first and last digits is: {new_number}")
```

```
Enter a number: 123456789
The number after swapping first and last digits is: 923456781
```

Print the following pattern using loop



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

In [103…

```python
# Butterfly Pattern in Python language of the numbers using for loop

rows, height, digits, space = None, None, None, None

print ("Enter the height of the pattern: ", end="")
height = int (input ())

print ("This the butterfly pattern:")

for rows in range (1, height):

    for digits in range (1, rows + 1):
        print ("*", end="")
    for space in range (1, (2 * (height - rows)) + 1):
        print (end=" ")

    for digits in range (rows, 0, -1):
        print ("*", end="")
    print ()

for rows in range (height, 0, -1):

    for digits in range (1, rows + 1):
        print ("*", end="")
    for space in range (1, (2 * (height - rows)) + 1):
        print (end=" ")

    for digits in range (rows, 0, -1):
        print ("*", end="")
    print ()
```

```
Enter the height of the pattern: 4
This the butterfly pattern:
*      *
**    **
***  ***
********
***  ***
**    **
*      *
```

In [81]:

```python
# Butterfly Pattern in Python language of the numbers using for loop

rows, height, digits, space = None, None, None, None

print ("Enter the height of the pattern: ", end="")
height = int (input ())

print ("\nThis the butterfly pattern:\n\n")

for rows in range (1, height):
    print (end="\t")
    for digits in range (1, rows + 1):
        print (digits, end="")
    for space in range (1, (2 * (height - rows)) + 1):
        print (end=" ")
    print (end="\b")
    for digits in range (rows, 0, -1):
        print (digits, end="")
    print (end="\n")

for rows in range (height, 0, -1):
    print (end="\t")
    for digits in range (1, rows + 1):
        print (digits, end="")
    for space in range (1, (2 * (height - rows)) + 1):
        print (end=" ")
    print (end="\b")
    for digits in range (rows, 0, -1):
        print (digits, end="")
    print (end="\n")
```

```
Enter the height of the pattern: 5

This the butterfly pattern:


        1       1
        12     21
        123   321
        1234 4321
        123454321
        1234 4321
        123   321
        12     21
        1       1
```

"Write a program to implement the calculator for the date of Easter.

The following algorithm computes the date for Easter Sunday for any year between 1900 to 2099. Ask the user to enter a year. Compute the following:

1. a = year % 19
2. b = year % 4
3. c = year % 7
4. d = (19 * a + 24) % 30
5. e = (2 *b* + *4* c + 6 * d + 5) % 7
6. dateofeaster = 22 + d + e

Special note: The algorithm can give a date in April. You will know that the date is in April if the calculation gives you an answer greater than 31. (You'll need to adjust) Also, if the year is one of four special years (1954, 1981, 2049, or 2076) then subtract 7 from the date.

Eg:

Input: Year : 2022

Expected Outcome: 2022-04-17 (i.e. 17th April 2022)"

In [2]:

```
# Get the year from the user
year = int(input("Enter a year between 1900 to 2099: "))

# Calculate the values using the given algorithm
a = year % 19
b = year % 4
c = year % 7
d = (19 * a + 24) % 30
e = (2 * b + 4 * c + 6 * d + 5) % 7
date_of_easter = 22 + d + e

# Check if the calculated date is in April
if date_of_easter > 31:
    # Adjust the date to be in April
    date_of_easter -= 31
    month = 4  # April
else:
    month = 3  # March

# Check if the year is one of the special years

if year==1954 or year==1981 or year==2049 or year== 2076:
    date_of_easter -= 7

# Display the date of Easter Sunday
print(f"The date of Easter Sunday in {year} is:{year}-{month:02d}-{date_of_easter:02d}
```

```
Enter a year between 1900 to 2099: 2022
The date of Easter Sunday in 2022 is:2022-04-17
```

"Write a Python program to compute the greatest common divisor (GCD) of two positive integers.

- The greatest common divisor (GCD) of two nonzero integers a and b is the greatest positive integer d such that d is a divisor of both a and b; that is, there are integers e and f such that a = de and b = df, and d is the largest such integer. The GCD of a and b is generally denoted gcd(a, b).
- For example, the greatest common factor of 15 and 10 is 5, since both the numbers can be divided by 5."

In [85]:

```
# Get two positive integers from the user
num1 = int(input("Enter the first positive integer: "))
num2 = int(input("Enter the second positive integer: "))

# Check if the input numbers are positive
if num1 <= 0 or num2 <= 0:
    print("Please enter positive integers.")
```

```
else:
    # Calculate GCD without using a function
    while num2 != 0:
        temp = num2
        num2 = num1 % num2
        num1 = temp

    # Display the GCD
    print(f"The Greatest Common Divisor (GCD) of the given numbers is: {num1}")
```

```
Enter the first positive integer: 120
Enter the second positive integer: 100
The Greatest Common Divisor (GCD) of the given numbers is: 20
```

"Write a python program that prompts the user to enter numbers and stops only when the use enter “QUIT” . After this print sum and average of the numbers, minimum and maximum number from given numbers entered by user.

Note: you are not allowed to use any built in structures like, list ,tuple etc. or any builtin function like min, max etc.

For Example: Input: 4,1,5,”QUIT”

Output:

Sum=10

Average=3.333

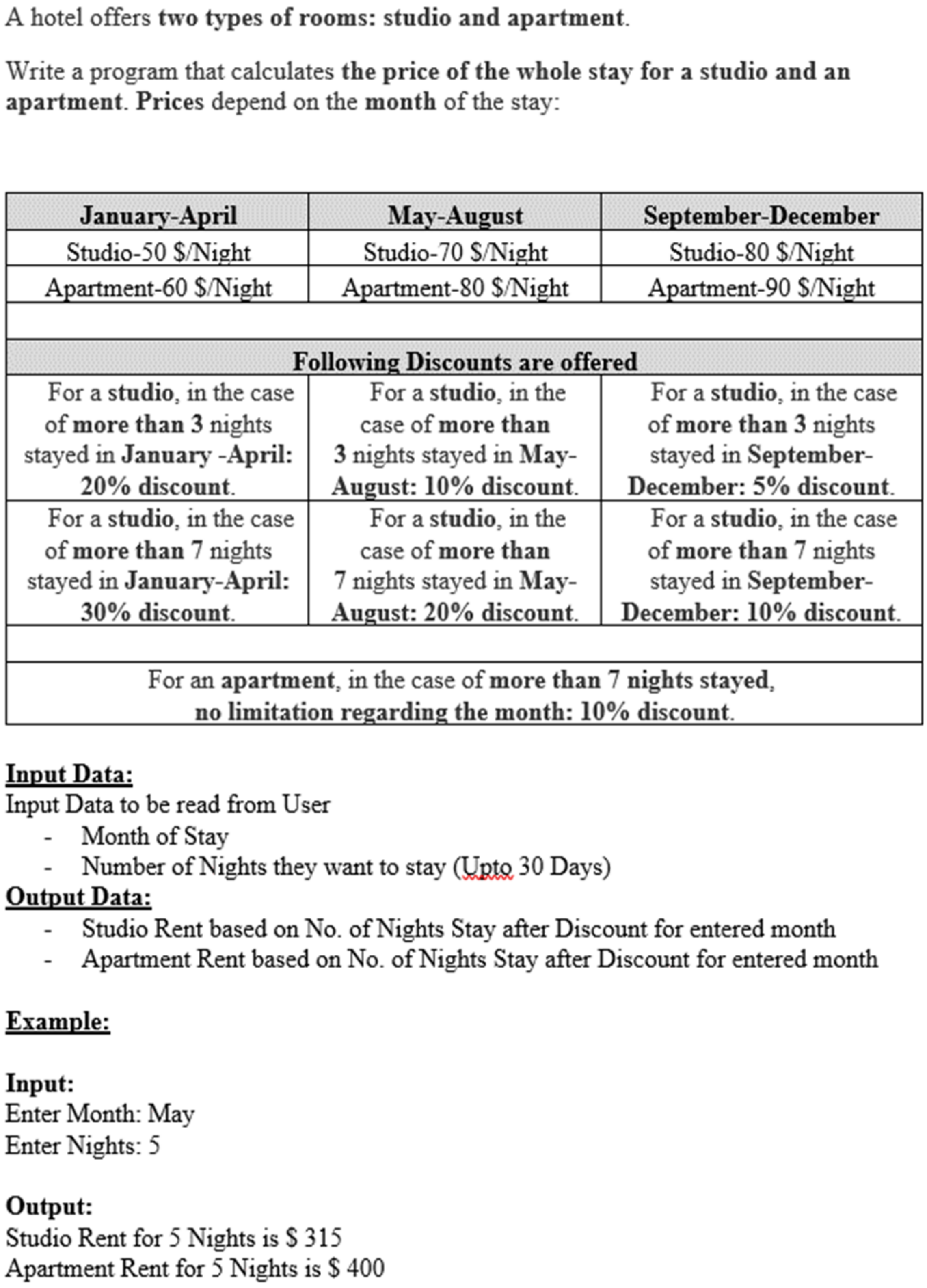
Minimum number=1

Maximum number=5"

In [86]:

```
sum = 0
count = 0
min_num = None
max_num = None
number = input("Enter a number (or QUIT to exit): ")

while number != "QUIT":
  number = int(number)
  sum += number
  count += 1

  if min_num is None or number < min_num:
    min_num = number

  if max_num is None or number > max_num:
    max_num = number

  number = input("Enter a number (or QUIT to exit): ")

if count > 0:
  average = sum / count
  print("Sum:", sum)
  print("Average:", average)
  print("Minimum number:", min_num)
  print("Maximum number:", max_num)
else:
  print("No numbers entered.")
```

```
Enter a number (or QUIT to exit): 4
Enter a number (or QUIT to exit): 1
Enter a number (or QUIT to exit): 5
Enter a number (or QUIT to exit): QUIT
Sum: 10
Average: 3.3333333333333335
Minimum number: 1
Maximum number: 5
```



A hotel offers **two types of rooms: studio and apartment**.

Write a program that calculates **the price of the whole stay for a studio and an apartment**. **Prices** depend on the **month** of the stay:

January-April	May-August	September-December
Studio-50 $/Night	Studio-70 $/Night	Studio-80 $/Night
Apartment-60 $/Night	Apartment-80 $/Night	Apartment-90 $/Night
Following Discounts are offered		
For a **studio**, in the case of **more than 3** nights stayed in **January -April: 20% discount**.	For a **studio**, in the case of **more than 3** nights stayed in **May-August: 10% discount**.	For a **studio**, in the case of **more than 3** nights stayed in **September-December: 5% discount**.
For a **studio**, in the case of **more than** 7 nights stayed in **January-April: 30% discount**.	For a **studio**, in the case of **more than** 7 nights stayed in **May-August: 20% discount**.	For a **studio**, in the case of **more than** 7 nights stayed in **September-December: 10% discount**.
For an **apartment**, in the case of **more than** 7 **nights stayed, no limitation regarding the month: 10% discount**.		

Input Data:

Input Data to be read from User

- Month of Stay
- Number of Nights they want to stay (Upto 30 Days)

Output Data:

- Studio Rent based on No. of Nights Stay after Discount for entered month
- Apartment Rent based on No. of Nights Stay after Discount for entered month

Example:

Input:

Enter Month: May

Enter Nights: 5

Output:

Studio Rent for 5 Nights is $ 315

Apartment Rent for 5 Nights is $ 400

In [92]:

```
# Get the month from the user
month = input("Enter the month: ")

# Get the number of nights from the user
num_nights = int(input("Enter the number of nights: "))

# Get the base price for the room type and month
Sbase_price = 0
if month == "january" or month == "february" or month == "march" or month == "april":
    Sbase_price = 50
elif month == "may" or month == "june" or month == "july" or month == "august":
    Sbase_price = 70
elif month == "september" or month == "october" or month == "november" or month == "de
    Sbase_price = 80
Abase_price=0
if month == "january" or month == "february" or month == "march" or month == "april":
    Abase_price = 60
elif month == "may" or month == "june" or month == "july" or month == "august":
    Abase_price = 80
elif month == "september" or month == "october" or month == "november" or month == "de
    Abase_price = 90

# Calculate the discount based on the number of nights
Sdiscount = 0
if num_nights >= 3 and (month == "january" or month == "february" or month == "march"
    Sdiscount = 20
elif num_nights >= 7 and (month == "january" or month == "february" or month == "march
    Sdiscount = 30
elif num_nights >= 3 and (month == "may" or month == "june" or month == "july" or mont
    Sdiscount = 10
elif num_nights >= 7 and (month == "may" or month == "june" or month == "july" or mont
    Sdiscount = 20
elif num_nights >= 3 and (month == "september" or month == "october" or month == "nove
    Sdiscount = 5
elif num_nights >= 7 and (month == "september" or month == "october" or month == "nove
    Sdiscount = 10
Adiscount = 0
if num_nights >= 7:
    Adiscount = 10

# Calculate the total price
Stotal_price = Sbase_price * num_nights - Sbase_price * num_nights * Sdiscount / 100
Atotal_price = Abase_price * num_nights - Abase_price * num_nights * Adiscount / 100

# Print the total price
print("The total price for the hotel room is:", Stotal_price)
print("The total price for the hotel room is:", Atotal_price)
```

```
Enter the month: may
Enter the number of nights: 5
The total price for the hotel room is: 315.0
The total price for the hotel room is: 400.0
```

"Write a program that enters a single digit integer number and produces all possible 6-digit numbers for which the product of their digits is equal to the entered number.

Example: ""number"" → 2

- 111112 → 1 *1* 1 *1* 1 * 2 = 2

- 111121 → 1 *1* 1 *1* 2 * 1 = 2

- 111211 → 1 *1* 1 *2* 1 * 1 = 2

- 112111 → 1 *1* 2 *1* 1 * 1 = 2

- 121111 → 1 *2* 1 *1* 1 * 1 = 2

- 211111 → 2 *1* 1 *1* 1 * 1 = 2"

In [88]:

```
# Get the single-digit integer from the user
number = int(input("Enter a single-digit integer: "))

# Check if the entered number is a single-digit integer
if 0 <= number <= 9:
    print(f"All possible 6-digit numbers for which the product of their digits is equa

    # Loop through all 6-digit numbers and check if their product of digits equals the
    for i in range(100000, 1000000):
        product = 1
        num_str = str(i)

        # Calculate the product of digits in the current 6-digit number
        for digit in num_str:
            product *= int(digit)

        # Check if the product matches the entered number and print the number if it m
        if product == number:
            print(i)
else:
    print("Please enter a valid single-digit integer.")
```

```
Enter a single-digit integer: 5
All possible 6-digit numbers for which the product of their digits is equal to 5:
111115
111151
111511
115111
151111
511111
```

"Write a Python program that prompts the user to enter numbers and stops only when the user enters “stop”. After this, print the minimum even, maximum even, average of even number, minimum odd, maximum odd, average of odd number from among all the numbers entered by the user.

- Note: You are not allowed to use any built-in structures like lists, tuples, etc. or any built-in functions like max, min, sum

Example: input and output

enter number or q for'stop:'-1

enter number or q for'stop:'-5

enter number or q for'stop:'9

enter number or q for'stop:'2

enter number or q for'stop:'4

enter number or q for'stop:'6

enter number or q for'stop:'stop

Output:

for even 6 2 4.0 (max, min, avg)

for odd 9 -5 1.0 (max, min, avg)"

In [102…

```
# Initialize variables for even numbers
sum_even = count_even = 0

# Initialize variables for odd numbers

sum_odd = count_odd = 0
count=0
# Get numbers from the user until "stop" is entered
while True:
    num = input("Enter a number or 'stop' to exit: ")

    # Check if user wants to stop
    if num == "stop":
        break
    num=int(num)

   # Check if the number is even or odd and update variables accordingly
    if num % 2 == 0:
        if count_even==0:
            max_even=num
            min_even=num
        else:
            # Update variables for even numbers
            if num < min_even:
                min_even = num
            if num > max_even:
                max_even = num
        sum_even += num
        count_even += 1
    else:
        if count_odd==0:
            max_odd=num
            min_odd=num
        else:
            # Update variables for odd numbers
            if num < min_odd:
                min_odd = num
            if num > max_odd:
                max_odd = num
        sum_odd += num
        count_odd += 1
```

```
# Calculate average for even and odd numbers
avg_even = sum_even / count_even if count_even != 0 else 0
avg_odd = sum_odd / count_odd if count_odd != 0 else 0

# Display the results for even and odd numbers
print(f"For even: {max_even if count_even else ''} {min_even if count_even else ''} {a
print(f"For odd: {max_odd if count_odd else ''} {min_odd if count_odd else ''} {avg_od
```

```
Enter a number or 'stop' to exit: 1
Enter a number or 'stop' to exit: 2
Enter a number or 'stop' to exit: 3
Enter a number or 'stop' to exit: 4
Enter a number or 'stop' to exit: 5
Enter a number or 'stop' to exit: 6
Enter a number or 'stop' to exit: 7
Enter a number or 'stop' to exit: 8
Enter a number or 'stop' to exit: 9
Enter a number or 'stop' to exit: 10
Enter a number or 'stop' to exit: stop
For even: 10 2 6.0 (max, min, avg)
For odd: 9 1 5.0 (max, min, avg)
```

In [ ]: