even or odd number

```
In [1]: print("amar")
        num=int(input("enter a number"))
        if num%2==0:
            print("num is even")
        else:
            print("num is odd")
```

```
amar
enter a number56
num is even
```

positive or negative number

```
In [2]: print("amar")
        num=int(input("enter an integer:"))
        if num>0:
            print("num is positive")
        elif num<0:
            print("num is odd")
        else:
            print("num is zero")
```

```
amar
enter an integer:45
num is positive
```

prime number

```
In [3]: print("amar")
        num=int(input("enter a number:"))
        count=0
        for i in range(1,num+1):
            if(num%i==0):
                count +=1
        if(count==2):
            print("the given number is prime")
        else:
            print("the given number is not prime")
```

```
amar
enter a number:34
the given number is not prime
```

pallindrome

```
In [4]: print("Amar")
        num=int(input("enter a number:"))
        num_str = str(num)
        if num_str == num_str[::-1]:
            print("num is pallindrome")
        else:
            print("num is not a pallindrome")
```

```
Amar
enter a number:53
num is not a pallindrome
```

sum of two numbers

```
In [5]: print("Amar")
        a=int(input("enter a num:"))
        b=int(input("enter a num:"))
        sum=a+b
        print(sum)
```

```
Amar
enter a num:6
enter a num:7
13
```

sum of two numbers using function

```
In [6]: print("Amar")
        def calculate_sum(num1,num2):
            return num1+num2
        num1=int(input("enter the number:"))
        num2=int(input("enter the number:"))
        sum=num1+num2
        print("sum:",sum)
```

```
Amar
enter the number:56
enter the number:53
sum: 109
```

maximum of two nubers

```
In [7]: print("Amar")
        num1=67
        num2=67
        result=max(num1,num2)
        print("maximum:",result)
```

```
Amar
maximum: 67
```

minimum of two numbers

In [9]:
```python
print("Amar")
num1=45
num2=78
result=min(num1,num2)
print("minimum:",result)
```

Amar
minimum: 45

fibonacci series

In [10]:
```python
print("Amar")
num=int(input("enter the fibonacci sequence length:"))
a=0
b=3
print("the fibonacci series of sequence",num,"is;")
print(a,b,end="")
for i in range(2,num):
    c=a+b
    print(c,end="")
    a=b
    b=c
```

Amar
enter the fibonacci sequence length:13
the fibonacci series of sequence 13 is;
0 33691524396310216526743Z

factorial number

In [11]:
```python
print("Amar")
n=int(input("enter the number:"))
f=1
if(n<0):
    print("not possible:")
elif(n==0):
    print("the factorial=1")
else:
    for i in range(1,n+1):
        f=f*i
print("factorial is:",f)
```

Amar
enter the number:56
factorial is: 71099858787804863451854045647463724949736497978881168458687447040
000000000000

reverse number

```
In [13]: print("Amar")
         num_str="5865hsfsydfdh"
         reversed_str=num_str[::-1]
         print("reversed number:",reversed_str)
```

```
Amar
reversed number: hdfdysfsh5685
```

swapping

```
In [14]: print("Amar")
         a=int(input("a="))
         b=int(input("b="))
         a,b=b,a
         print("after swapping:")
         print("a:",a)
         print("b:",b)
```

```
Amar
a=45
b=98
after swapping:
a: 98
b: 45
```

gcd of two numbers

```
In [15]: print("Amar")
         import math
         num1=int(input("enter a number:"))
         num2=int(input("enter a number:"))
         result=math.gcd(num1,num2)
         print("result:",result)
```

```
Amar
enter a number:67
enter a number:98
result: 1
```

random numbers

In [16]:
```python
print("Amar")
import random
number=random.randint(1,10)
guess=0
while guess!=number:
    guess=int(input("guess a number"))
    if guess<number:
        print("guess a higher number")
    elif guess>number:
        print("guess a lower number")
    else:
        print("you guessed the correct number",number)
```

```
Amar
guess a number7
guess a higher number
guess a number9
guess a lower number
guess a number8
you guessed the correct number 8
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: