Program to find the factorial of a given number

* The factorial of a number is the product of all the integers from 1 to that number. For example, the factorial of 6 (denoted as 6!) is 1\*2\*3\*4\*5\*6 = 720.
* Factorial is not defined for negative numbers and the factorial of zero is one, 0! = 1.

**Sol 1: Using for loop**

**Version 1**

n=int(input("enter a number"))

fact=1

if n < 0:

print("fact does not exists")

elif n == 0:

print("fact is 1")

else:

for i in range(2,n+1):

fact\*=i

print(fact)

**Version 2**

n=int(input("enter a number"))

fact=1

if n < 0:

print("fact does not exists")

elif n == 0:

print("fact is 1")

else:

for i in range(n,1,-1):

fact\*=i

print(fact)

**Sol 2: Using while loop**

**Version 1**

n=int(input("enter a number"))

fact=1

i=1

if n < 0:

print("fact does not exists")

elif n == 0:

print("fact is 1")

else:

while i<=n:

fact\*=i

i=i+1

print(fact)

**Version 2**

n=int(input("enter a number"))

fact=1

if n < 0:

print("fact does not exists")

elif n == 0:

print("fact is 1")

else:

while n:

fact\*=n

n=n-1

print(fact)

**Version 3**

n=int(input("enter a number"))

fact=1

if n < 0:

print("fact does not exists")

elif n == 0:

print("fact is 1")

else:

while n >= 1:

fact\*=n

n=n-1

print(fact)

**Program to find the factorial of a random integer**

**While loop**

from random import randint

n=randint(5,10)

print(n)

fact=1

while n:

fact\*=n

n=n-1

print(fact)

**for loop**

from random import randint

n=randint(5,10)

print(n)

fact=1

for i in range(1,n+1):

fact\*=i

print(fact)

or

from random import randint

n=randint(5,10)

print(n)

fact=1

for i in range(n,0,-1):

fact\*=i

print(fact)

**Program to find the factorial of a random integer between p and q**

**While Loop**

from random import randint

p=int(input("enter the start index value"))

q=int(input("enter the stop index value"))

n=randint(p,q)

print(n)

fact=1

while n:

fact\*=n

n=n-1

print(fact)

**for Loop**

from random import randint

p=int(input("enter the start index value"))

q=int(input("enter the stop index value"))

n=randint(p,q)

print(n)

fact=1

for i in range(1,n+1):

fact\*=i

print(fact)

Note : randrange() can be used

What's the difference between randrange and randint?

There is one slight difference when used with just two parameters. randint(x,y) will return a value >= x and <= y, while randrange(x,y) will return a value >=x and < y (n.b. not less than or equal to y)

There is a bigger difference if you use the additional parameter that randrange can take ... randrange(start, stop, step). randrange will return a random number that is contained in the list. So randrange(0,30,5) would return ONE of the numbers in [0, 5, 10, 15, 20, 25].

**While Loop**

from random import randrange

p=int(input("enter the start index value"))

q=int(input("enter the stop index value"))

n=randrange(p,q+1)

print(n)

fact=1

while n:

fact\*=n

n=n-1

print(fact)

**for Loop**

from random import randrange

p=int(input("enter the start index value"))

q=int(input("enter the stop index value"))

n=randrange(p,q+1)

print(n)

fact=1

for i in range(n,0,-1):

fact\*=i

print(fact)

**Program find factorial of an integer N where N is an random even number selected in 10 <= N < 100**

**While loop**

from random import randrange

n=randrange(10,100, 2)

print(n)

fact=1

while n:

fact\*=n

n=n-1

print(fact)

or

from random import randrange

p=int(input("enter the start index value"))

q=int(input("enter the stop index value"))

n=randrange(p,q,2)

print(n)

fact=1

while n:

fact\*=n

n=n-1

print(fact)

**for loop**

from random import randrange

n=randrange(10,100, 2)

print(n)

fact=1

for i in range(1,n+1):

fact\*=i

print(fact)

or

from random import randrange

p=int(input("enter the start index value"))

q=int(input("enter the stop index value"))

n=randrange(p,q,2)

print(n)

fact=1

for i in range(1,n+1):

fact\*=i

print(fact)

**Program to find sum of factorial of even and odd numbers from 1 to 100**

sen=0

son=0

fact=1

for i in range(1,6):

fact\*=i

if i%2:

son+=fact

else:

sen+=fact

print(sen,son)

sen=0

son=0

fact=1

n=5

i=1

while i<=n:

fact=fact\*i

if i%2==0:

sen=sen+fact

else:

son=son+fact

i=i+1

print(sen,son)

**Program to generate factorial series from 1 to N**

**Using While loop**

lower=1

upper=int(input("enter the ending index"))

while lower<=upper:

n=lower

fact=1

while n:

fact\*=n

n=n-1

print(fact)

lower=lower+1

**using For loop**

lower=1

upper=int(input("enter the ending index"))

fact=1

for n in range(lower,upper+1):

fact\*=n

n=n-1

print(fact)

**Program to generate factorial series from 1 to random integer**

**Using While loop**

from random import randint

lower=1

upper=randint(5,10)

while lower<=upper:

n=lower

fact=1

while n:

fact\*=n

n=n-1

print(fact)

lower=lower+1

**using For loop**

from random import randint

lower=1

upper=randint(5,10)

fact=1

for n in range(lower,upper+1):

fact\*=n

n=n-1

print(fact)

**Program to generate factorial series from 1 to N, where N is an random even number selected in 10 <= N < 100**

**Using While loop**

from random import randrange

lower=1

upper= randrange(10,100, 2)

while lower<=upper:

n=lower

fact=1

while n:

fact\*=n

n=n-1

print(fact)

lower=lower+1

**using For loop**

from random import randrange

lower=1

upper= randrange(10,100, 2)

fact=1

for n in range(lower,upper+1):

fact\*=n

n=n-1

print(fact)

Program to generate factorial series from 1 to N, where N is an random odd number selected in 11 <= N < 100

**Using While loop**

from random import randrange

lower=1

upper= randrange(11,100, 3)

while lower<=upper:

n=lower

fact=1

while n:

fact\*=n

n=n-1

print(fact)

lower=lower+1

**using For loop**

from random import randrange

lower=1

upper= randrange(11,100, 3)

fact=1

for n in range(lower,upper+1):

fact\*=n

n=n-1

print(fact)

**Program to generate Fibonacci sequence**

* A Fibonacci sequence is the integer sequence of 0, 1, 1, 2, 3, 5, 8....
* The first two terms are 0 and 1.
* All other terms are obtained by adding the preceding two terms.
* This means to say the nth term is the sum of (n-1)th and (n-2)th term.

Solution 1: Using while loop

Version 1

nterms = int(input("How many terms? "))

# first two terms

n1 = 0

n2 = 1

count = 2

# check if the number of terms is valid

if nterms <= 0:

print("Plese enter a positive integer")

elif nterms == 1:

print("Fibonacci sequence:")

print(n1)

else:

print("Fibonacci sequence:")

print(n1,",",n2,end=', ')

while count < nterms:

nth = n1 + n2

print(nth,end=' , ')

# update values

n1 = n2

n2 = nth

count += 1

Version 2

#This program calculates the Fibonacci sequence

max\_count = int(input("How many terms? "))

n1 = 0

n2 = 1

count = 0

while count < max\_count:

count = count + 1

#we need to keep track of a since we change it

old\_n1 = n1

old\_n2 = n2

n1 = old\_n2

n2 = old\_n1 + old\_n2

#Notice that the , at the end of a print statement keeps it

# from switching to a new line

print (old\_n1,end=",")

Solution 2: Using for loop

Version 1

nterms = int(input("How many terms? "))

n1 = 0

n2 = 1

if nterms <= 0:

print("Plese enter a positive integer")

elif nterms == 1:

print("Fibonacci sequence:")

print(n1)

else:

print("Fibonacci sequence:")

for i in range(0,nterms):

print(n1)

n1,n2=n2,n1+n2

**Program to generate Fibonacci sequence from 0 to random integer or generate Fibonacci sequence up to a random integer between 5 and 10**

from random import randint

nterms=randint(5,10)

print(nterms)

# first two terms

n1 = 0

n2 = 1

count = 2

# check if the number of terms is valid

if nterms <= 0:

print("Plese enter a positive integer")

elif nterms == 1:

print("Fibonacci sequence:")

print(n1)

else:

print("Fibonacci sequence:")

print(n1,",",n2,end=', ')

while count < nterms:

nth = n1 + n2

print(nth,end=' , ')

# update values

n1 = n2

n2 = nth

count += 1

**Note: similarly try with for loop and other variants like taking p and q as range, try randrange() and randrange() with limits p and q similar to factorial program / problem above.**

**Program to generate Fibonacci sequence upto N and compute sum**

nterms = int(input("How many terms? "))

n1 = 0

n2 = 1

sum=1

if nterms <= 0:

print("Plese enter a positive integer")

elif nterms == 1:

print("Fibonacci sequence:")

print(n1)

else:

print("Fibonacci sequence:")

for i in range(0,nterms):

print(n1)

sum+=n1

n1,n2=n2,n1+n2

print("sum is",sum)

**Note: similarly try with while loop and other variants like randint(), taking p and q as range, try randrange() and randrange() with limits p and q similar to factorial program / problem above.**

**Program to print digits of a given integer**

n=int(input("Enter an integer"))

while n:

digits=n%10

n=n//10

print(digits)

n=int(input("Enter an integer"))

while n:

r,n=n%10,n//10

print(r)

n=int(input("Enter an integer"))

while n:

n,r=divmod(n,10)

print(r)

n=int(input("Enter an integer"))

for digits in range(1,len(str(n))+1):

print(digits)

**Program to print odd digits of a given integer**

n=int(input("Enter an integer"))

while n:

digits=n%10

n=n//10

if digits%2:

print(digits)

n=int(input("Enter an integer"))

while n:

r,n=n%10,n//10

if r%2:

print(r)

n=int(input("Enter an integer"))

while n:

n,r=divmod(n,10)

if r%2:

print(r)

**Program to print even digits of a given integer**

n=int(input("Enter an integer"))

while n:

digits=n%10

n=n//10

if digits%2==0:

print(digits)

n=int(input("Enter an integer"))

while n:

r,n=n%10,n//10

if r%2==0:

print(r)

n=int(input("Enter an integer"))

while n:

n,r=divmod(n,10)

if r%2==0:

print(r)

**Program to print sum of odd digits of a given integer**

n=int(input("Enter an integer"))

s=0

while n:

digits=n%10

n=n//10

if digits%2:

s+=digits

print(s)

n=int(input("Enter an integer"))

s=0

while n:

r,n=n%10,n//10

if r%2:

s=s+r

print(s)

n=int(input("Enter an integer"))

s=0

while n:

n,r=divmod(n,10)

if r%2:

s=s+r

print(s)

**Program to print even digits of a given integer**

n=int(input("Enter an integer"))

s=0

while n:

digits=n%10

n=n//10

if digits%2==0:

s+=digits

print(s)

n=int(input("Enter an integer"))

s=0

while n:

r,n=n%10,n//10

if r%2==0:

s=s+r

print(s)

n=int(input("Enter an integer"))

s=0

while n:

n,r=divmod(n,10)

if r%2==0:

s=s+r

print(s)

**Program to print sum of odd and even digits of a given integer**

n=int(input("Enter an integer"))

so=0

se=0

while n:

digits=n%10

n=n//10

if digits%2:

so+=digits

else:

so+=digits

print(so,se)

n=int(input("Enter an integer"))

so=0

se=0

while n:

r,n=n%10,n//10

if r%2==0:

se=se+r

else:

so=so+r

print(se)

print(so)

n=int(input("Enter an integer"))

so=0

se=0

while n:

n,r=divmod(n,10)

if r%2:

se=se+r

else:

so=so+r

print(se)

print(so)

**Program to Find Sum of digits of a Given Integer**

Solution 1: while loop

**Version 1**

s=0

n=int(input("enter the number"))

while n:

s += n%10

n //= 10 #floor div

print(s)

**Version 2**

s=0

n=int(input("enter an integer"))

while n:

n,r=divmod(n,10)

s += r

print(s)

**Version 3**

r=0

n=int(input("enter an integer"))

while n:

r,n=r+n%10,n//10

print(r)

**Version 4**

r=0

s=0

n=int(input("enter an integer"))

while n>0:

r=n%10

n=n//10

s=s+r

print(s)

Version 5

n=int(input("enter the no"))

s=0

r=0

while n:

r=n%10

s=s+r

n=n//10

print(s)

Solution 2: Using For Loop

n=input("enter the integer")

s=0

for digits in n:

s+=int(digits)

print(s)

**Solution 3: Using Sum function and for loop**

num=input("enter a number")

print(sum(int(digit) for digit in num))

**Program to find sum of first n natural numbers**

**While loop**

nstrt=0

sum=0

nstp=int(input("enter a value to stop"))

while nstrt <= nstp:

sum+=nstrt

nstrt=nstrt+1

print("sum={:0}".format(sum))

**for loop**

nstrt=0

sum=0

nstp=int(input("enter a value to stop"))

for num in range(nstrt,nstp+1):

sum+=num

print("sum={:0}".format(sum))

**do…while simulation using while True:**

nstrt=0

sum=0

nstp=int(input("enter a value to stop"))

while True:

sum+=nstrt

if nstrt==nstp:

break

else:

nstrt =nstrt+1

print("sum={:0}".format(sum))

or

nstrt=0

sum=0

nstp=int(input("enter a value to stop"))

while True:

sum+=nstrt

if nstrt==nstp:

break

nstrt =nstrt+1

print("sum={:0}".format(sum))

**Program to find sum of first n odd natural numbers**

**While loop**

nstrt=0

osum=0

nstp=int(input("enter a value to stop"))

while nstrt <= nstp:

if nstrt%2:

osum+=nstrt

nstrt=nstrt+1

print("sum={:0}".format(osum))

**for loop**

nstrt=0

osum=0

nstp=int(input("enter a value to stop"))

for num in range(nstrt,nstp+1):

if nstrt%2:

osum+= num

print("sum={:0}".format(sum))

**do…while simulation using while True:**

nstrt=0

sum=0

nstp=int(input("enter a value to stop"))

while True:

if nstrt%2:

sum+=nstrt

if nstrt==nstp:

break

nstrt =nstrt+1

print("sum={:0}".format(sum))

or

nstrt=0

sum=0

nstp=int(input("enter a value to stop"))

while True:

if nstrt%2:

sum+=nstrt

if nstrt==nstp:

break

else:

nstrt =nstrt+1

print("sum={:0}".format(sum))

**Program to find sum of first n even natural numbers**

**While loop**

nstrt=0

esum=0

nstp=int(input("enter a value to stop"))

while nstrt <= nstp:

if nstrt%2==0:

esum+=nstrt

nstrt=nstrt+1

print("sum={:0}".format(esum))

or

nstrt=0

esum=0

nstp=int(input("enter a value to stop"))

while nstrt <= nstp:

esum+=nstrt

nstrt=nstrt+2

print("sum={:0}".format(esum))

**for loop**

nstrt=0

esum=0

nstp=int(input("enter a value to stop"))

for num in range(nstrt,nstp+1,2):

esum+= num

print("sum={:0}".format(esum))

**Program to find the sum of squares of all positive even numbers up to 100**

**while loop**

nstrt=2

esum=0

nstp=100

while nstrt <= nstp:

esum+=nstrt\*\*2

nstrt=nstrt+2

print("sum={:0}".format(esum))

**for loop**

nstrt=0

esum=0

nstp=100

for num in range(nstrt,nstp+1,2):

esum+= num\*\*2

print("sum={:0}".format(esum))

**Write a program to compute the average of the ten numbers 1, 4, 9, ..., 81, 100.**

**while loop**

nstrt=1

sum=0

nstp=10

while nstrt <= nstp:

sum+=nstrt\*\*2

nstrt=nstrt+1

print("sum={:0}".format(sum))

print(("Avg={:}".format(sum/10))

**for loop**

nstrt=1

sum=0

nstp=10

for num in range(nstrt,nstp+1):

esum+= num\*\*2

print("sum={:0}".format(sum))

print(("Avg={:}".format(sum/10))

**Program to compute sum of numbers entered until encountering a negative number i.e. sentinel value**

s=0

num=input("Enter a number (neg no to stop)")

num=eval(num)

while num > 0:

s=s+num

num=input("Enter a number (neg no to stop)")

num=eval(num)

print(s)

**Program to compute sum of numbers entered until encountering EOF / <enter> key i.e. sentinel value**

s=0

num=input("Enter a number (<enter> to stop)")

while num != "":

num=eval(num)

s=s+num

num=input("Enter a number (<enter> to stop)")

print(s)

**Program to find whether number is Armstrong number or not**

If the sum of the number of their own digits to the power of the number of digits is equal to that number then that number is called as Armstrong number. In other words an Armstrong number is an n-digit number that is equal to the sum of the nth powers of its digits.

For example : let number be 370 then 370 is of 3 digit number. so if 3^3 + 7^3 + 0^3 = 370 then the number is called as Armstrong number. All one digit number are Armstrong number i.e 1,2,3,4,5,6,7,8,9.

Their exist no 2 digit number which is Armstrong number. For 3 digit 153, 370, 371 and 407 and For 4 digit 1634, 8208 and 9474 are the example of Armstrong number.

**Solution1 : While loop**

**Version 1**

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

nod=len(str(num))

sum = 0

temp = num

while temp > 0: #or just write while temp:

digit = temp % 10

sum += digit \*\* nod

temp //= 10

if num == sum:

print(num,"is an Armstrong number")

else:

print(num,"is not an Armstrong number")

**Version 2**

sum = 0

no = int(input("Enter the the number to check it is armstrong on or not :"))

pow\_no = len(str(no))

check = no

while no:

sum = sum +((no%10)\*\*pow\_no)

no = no // 10

if check == sum:

print ("%s is Armstrong"%check)

else:

print ("%s is not Armstrong"%check)

**Solution 2 : for loop**

**Version 1**

num = input("Enter a number: ")

nod=len(num)

sum = 0

for t in num:

sum += int(t) \*\* nod

if int(num) == sum:

print(num,"is an Armstrong number")

else:

print(num,"is not an Armstrong number")

**Version 2**

sum=0

no = int(input("Enter the the number to check it is armstrong on or not :"))

pow\_no = len(str(no))

check = no

for i in str(no):

sum = sum + (int(i)\*\*pow\_no)

if check == sum:

print ("%s is Armstrong"%check)

else:

print ("%s is not Armstrong"%check)

**Program to check whether given number is prime or not**

A positive integer greater than 1 which has no other factors except 1 and the number itself is called a prime number. 2, 3, 5, 7 etc. are prime numbers as they do not have any other factors. But 6 is not prime (it is composite) since, **2 x 3 = 6**.

# Python program to check if the input number is prime or not

# take input from the user

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

# prime numbers are greater than 1

if num > 1:

# check for factors

for i in range(2,num):

if (num % i) == 0:

print(num,"is not a prime number")

print(i,"times",num//i,"is",num)

break

else:

print(num,"is a prime number")

# if input number is less than

# or equal to 1, it is not prime

else:

print(num,"is not a prime number")

In this program, user is asked to enter a number and this program check whether that number is prime or not. Numbers less than or equal to 1 are not prime numbers. Hence, we only proceed if the **num** is greater than 1. We check if **num** is exactly divisible by any number from 2 to **num** - 1. If we find a factor in that range, the number is not prime. Else the number is prime.

We can decrease the range of numbers where we look for factors. In the above program, our search range is from 2 to **num** - 1. We could have used the range, [2, **num** / 2] or [2, **num** \*\* 0.5]. The later range is based on the fact that a composite number must have a factor less than square root of that number. Otherwise the number is prime.

**Python program to check if the input number is prime or no**

# take input from the user

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

# prime numbers are greater than 1

if num > 1:

# check for factors

for i in range(2,num//2): #or for i in range(2,num\*\*0.5):

if (num % i) == 0:

print(num,"is not a prime number")

print(i,"times",num//i,"is",num)

break

else:

print(num,"is a prime number")

# if input number is less than

# or equal to 1, it is not prime

else:

print(num,"is not a prime number")

**Write a Program to Print all Prime Numbers in an Interval**

lower = int(input("Enter lower range: "))

upper = int(input("Enter upper range: "))

for num in range(lower,upper + 1):

# prime numbers are greater than 1

if num > 1:

for i in range(2,num): # range(2,int(num\*\*0.5))

# range(2,num//2)

if (num % i) == 0:

break

else:

print(num)

**Smallest of three integers without comparison operators**

Take a counter variable c and initialize it with 0. In a loop, repeatedly subtract n1, n2 and n3 by 1 and increment c. The number which becomes 0 first is the smallest. After the loop terminates, c will hold the minimum of 3.

n1=int(input("enter first number"))

n2=int(input("enter second number"))

n3=int(input("enter third number"))

c = 0;

while n1 and n2 and n3 :

n1=n1-1

n2=n2-1

n3=n3-1

c=c+1

print(c)