############ PROGRAM 1 ########################################

#python program to print all numbers in which each no has  has only two factors

n=int(input("Enter no: "))

a=[]

for i in range(2,n):

    count=1

    for j in range(1,i):

        if i%j==0:

            count+=1

    if count==2:

        a.append(i)

    else:

        continue

print(a)

C:\Users\user\Desktop\python examples>codeprac.py

Enter no: 10

[2, 3, 5, 7]

##############PROGRAM 2 ############################################

**#1 .binary search to find prime no 73 in a given list**

a=[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]

b=73

def binary\_search(a,b):

low=0

high=len(a)-1

mid=0

while low<=high:

mid=(high+low)//2

if a[mid]<b:

low=mid+1

elif a[mid]>b:

high=mid-1

else:

return mid

return -1

result binary\_search(a,b)

if result!=-1:

print("Element is present at index: "+str(result))

else:

print("Element is not present")

**2.linear search to find prime no 73 in a given list**

a=[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]

b=int(input("Enter no :"))

for i in a:

if i==b:

print("Element is present: "+str(b))

break

else:

print("No is not found:")

**3.#program to print factorial no( 4!=4\*3\*2\*1)**

n=int(input("Enter a no to find factorial: "))

def fact(n):

if n==1:

return n

else:

return n\*fact(n-1)

print("The factorial is:"+str(fact(n)))

##################################################################################

**#4.finding maximum product from a array through a quadraple(considering four elements:)**

def arra(arr,n):

if n<4:

return -1

max\_pro=n

for i in range(n-3):

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

for k in range(j+1,n-1):

for l in range(k+1,n):

max\_pro=max(max\_pro,arr[i]\*arr[j]\*arr[k]\*arr[l])

#prod=prod\*arr[i]\*arr[j]\*arr[k]\*arr[l]

return max\_pro

if \_\_name\_\_ == "\_\_main\_\_":

n=int(input())

arr=list(map(int,input().split(" ")))[:n]

print(arr)

ans=arra(arr,n)

if ans==-1:

print("No quadraple exists:")

else:

print("Max product is: "+str(ans))

**OUTPUT:**

C:\Users\user\Desktop\python examples>check.py

6

2 5 -1 -5 6 8

[2, 5, -1, -5, 6, 8]

Max product is: 480

**#5python program to find common factor of two numbers(HCF and also counting of factors)**

def gcd(a,b):

if (b==0):

return a

else:

return gcd(b,a%b)

a=int(input("Enter first no: "))

b=int(input("Enter second number: "))

print(gcd(a,b))

#if(a>0 and a<(10\*\*12+1) and b>=1 and b<(10\*\*12+1)):

count=0

for i in range(1,gcd(a,b)+1):

if a%i==0 and b%i==0:

count=count+1

print(count)

############################################################################

**#6problem of finding divide(IF the multiplication of no is divisible by sum of numbers then print yeah otherwise no)**

n=int(input("Enter the array size: "))

arr=[]

for i in range(n):

arr.append(int(input()))

print(arr)

sum=0

mul=1

for i in arr:

mul=mul\*i

sum=sum+i

print(sum)

print(mul)

if mul%sum==0:

print("yeah")

else:

print("nah")

#########################################################################

**#7 python program to remove duplicate char from string(A character appearing more than one time is duplicate character)**

n=input("Enter a string: ")

s=''

for ele in n:

if ele in s:

pass

else:

s=s+ele

print(s)

**#8 python program for printing hailstone sequence(The Hailstone Sequence of numbers can be started from a given positive integer N.**

**The remaining terms of the sequence can be evaluated by applying the following rules:**

1. **If N = 1, then the sequence terminates.**
2. **If N is even, then the next term of the sequence is N / 2.**
3. **If N is odd, then the next term in the sequence is 3 × N + 1.**

n=int(input("Enter no"))

a=[]

while n!=1:

a.append(n)

if n%2==0:

n=n//2

else:

n=n\*3+1

a.append(1)

for i in a:

print(i,end=" ")

print(len(a))

**OUTPUT:**

C:\Users\user\Desktop\python examples>comp.py

Enter first no: 3

Enter second number: 9

3

2

Enter the array size: 5

4

6

7

8

4

[4, 6, 7, 8, 4]

29

5376

nah

Enter a string: wertrtyuewert

wertyu

Enter no 4

4 2 1 3

#############################################################################

**#9 program for printing minimum index array element**

arr=[1,2,3,2,1]

n=len(arr)

arr.sort()

c=[]

for i in range(0,n):

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

if arr[i]==arr[j]:

c.append(arr[j])

c.sort()

print(min(c))

**#10 program to check if subarry sum is equal to zero**

a=[4,2,-3,1,8]

n=len(a)

sum=0

for i in range(n):

sum=sum+a[i]

if sum==0 or sum in a:

print("Yes")

**#11 strings are pangram or not(pangram : if all the alphabets of English letters are present in a string then the string is pangram)**

s1="the quick brown fox jumps over a lazy dog"

s2="abcdefghijklmnopqrstuvwxyz"

for ch in s2:

if ch in s1:

pass

else:

print("Not pangram")

**#12:code for the saga of the missing coin(A person came with no of coins and somehow a coin lost .He told that he has all the even no of coins. Now find out what no was the missing coin )**

n=len(a)

a=[5,7,2,7,5,2,5]

print(a)

count=0

for i in range(0,n):

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

if a[i]==a[j]:

count+=1

if count%2==0:

pass

else:

print("Value is: "+str(a[i]))

**#13:python program to check if a matrix is identity matrix[An identity matrix is a square matrix in which all the elements of principal diagonals are one, and all other elements are zeros)**

n=int(input("Enter the size of array: "))

mat=[]

print("Enter the values: ")

for i in range(n):

a=[]

for j in range(n):

a.append(int(input()))

mat.append(a)

for i in range(n):

for j in range(n):

print(mat[i][j],end=" ")

print()

def isIde(mat,n):

for i in range(n):

for j in range(n):

if i==j and mat[i][j]!=1:

return False

elif i!=j and mat[i][j]!=0:

return False

return True

if isIde(mat,n):

print("Yes")

else:

print("No")

**OUTPUT:**

C:\Users\user\Desktop\python examples>condr.py

1

Yes

Yes

[5, 7, 2, 7, 5, 2, 5]

Value is: 5

Enter the size of array: 2

Enter the values:

1

0

0

1

1 0

0 1

Yes

**####### CONVERSIONS QUESTION##################**

**#14 DECIMAL NO TO OCTAL CONVERSION**

n=int(input("Enter DEcimal vlaue: "))

def dctoOctal(n):

octal,r=0,0

i=1

while(n!=0):

r=n%8

octal=octal+r\*i

n=n//8

i=i\*10

return octal

print(dctoOctal(n))

**#OCTALTO DECIMAL CONVERSION**

import math

n=int(input("Enter octal: "))

def octtodec(n):

decimal,i=0,0

while(n!=0):

decimal=decimal+(n%10)\*math.pow(8,i)

i+=1

n=n//10

return decimal

print(octtodec(n))

**#15 python program to convert a number from binary to octal**

import math

def bintooct(binary):

octal,decimal,i=0,0,0

while(binary!=0):

decimal=decimal+(binary%10)\*math.pow(2,i)

i=i+1

binary=binary//10

i=1

while(decimal!=0):

octal=octal+(decimal%8)\*i

decimal=int(decimal//8)

i\*=10

return octal

return decimal

binary=int(input("Enter binary no: "))

print(bintooct(binary))

**#16 OCTALTOBINARY CONVERSION**

def octaltobinary(octal):

decimal,i,binary=0,0,0

while(octal!=0):

decimal=decimal+(octal%10)\*math.pow(8,i)

i+=1

octal=octal//10

i=1

while(decimal!=0):

binary=binary+(decimal%2)\*i

decimal=decimal//2

i=i\*10

return binary

octal=int(input("Enter octal value: "))

print(octaltobinary(octal))

**#17 hexadecimal to decimal**

hexa=input("Enter hexadecimal no: ")

dec=int(hexa,16)

print("hexadecimal to decimal: "+str(dec))

**#18 binto hexadecimal**

b=int(input("Enter binary value"))

temp=int(b,2)

print("binary to hexadecimal : "+str(temp))

**#19 decimal to binary conversion**

def decimalToBin(n):

if n>1:

decimalToBin(n//2)

print(n%2,end=" ")

if \_\_name\_\_ == "\_\_main\_\_":

decimalToBin(8)

print("\n")

**#20 binary to decimal conversion**

def bintodecimal(binary):

decimal,i=0,0

while(binary!=0):

r=binary%10

decimal=decimal+r\*pow(2,i)

binary=binary//10

i+=1

print(decimal)

if \_\_name\_\_ == "\_\_main\_\_":

bintodecimal(1000)

##########################################################################

**#21 python program for calculating hypotenuse of a triangle**

import math

P =float(input("Enter value of perpendicular: "))

B =float(input("Enter value of Base: "))

H=math.sqrt(pow(P,2)+pow(B,2))

print(H)

**#22 printing sum of all prime no in a given range**

a=int(input("Enter lower bound: "))

b=int(input("Enter lower bound: "))

sum=0

for c in range(a,b):

for i in range(2,c):

if c%i==0:

break

else:

print(c,end=" ")

sum=sum+c

print(sum)

print('\n')

**#################SERIES QUESTION##############**

**#find 15th term in the series**

n=int(input("Enter the nth term: "))

a=0

b=0

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

if i%2==0:

a=a+6

elif i%2!=0:

b=b+7

if (n%2!=0):

print(n,b-7)

else:

print(n,a-6)

**#1,1,2,3,4,9,8,27,16,81,32,243**

n1=int(input("Enter a number: "))

a=1

b=1

for i in range(1,n1+1):

if i%2==0:

b=b\*3

else:

a=a\*2

if(n1%2!=0):

print(n1,a/2)

else:

print(n1,b/3)

**#0,0,2,1,4,2,6,3,8,4,10**

n2=int(input("Enter the term: "))

a=0

b=0

for i in range(1,n2+1):

if i%2!=0:

a=a+2

else:

b=b+1

if (n2%2!=0):

print(a-2)

else:

print(b-1)

##################################################################################

**#24 replacing string with vowels and consonants and changing uppercase to lower and lower to uppercase letters in a string:**

s1='Hii I love my parents'

s2="Hello MySelf"

s3="Hii I am String"

vowels={'a','e','i','o','u','A','I','E','O','U'}

consonants={'b','l','H','M','y','s','f'}

for x in s1:

if x in vowels:

s1=s1.replace(x,'%')

print(s1)

for x in s2:

if x in consonants:

s2=s2.replace(x,'&')

print(s2)

newsr=""

for i in range(len(s3)):

if s3[i].islower():

newsr=newsr+s3[i].upper()

elif s3[i].isupper():

newsr=newsr+s3[i].lower()

else:

newsr=newsr+s3[i]

print(newsr)

**#print(s3.swapcase())(inbuilt function of python)**

* **#25 python program to check if a no is strong no or not(Strong number is a special number whose sum of factorial of digits is equal to the original number. For example: 145 is strong number. Since, 1! + 4! + 5! = 145)**

n=int(input("Enter a no to check if a no is strong no or not: "))

fact=1

sum=0

t=n

while n!=0:

r=n%10

i=1

while(i<=r):

fact=fact\*i

i=i+1

sum=sum+fact

n=n//10

if sum==t:

print("is a strong no"+str(t))

else:

print("Not a strong no")

**#26 Finding lcm of numbers**

def gcd(a,b):

if a==0:

return b

else:

return gcd(b%a,a)

def lcm(a,b):

return (a/gcd(a,b))\*b

print(lcm(15,20))

**# 27 check if a no is perfect no or not (A perfect number is a positive integer that is equal to the sum of its positive divisors, excluding the number itself.For instance, 6 has divisors 1, 2 and 3 (excluding itself), and 1 + 2 + 3 = 6, so 6 is a perfect number.)**

n1=int(input("Enter a no: "))

p=n1

su=0

def perfect\_no(n1):

for x in range(n1):

if n1%x==0:

su+=x

return su

if su==p:

print("Perfect no")

else:

print("no")

**# 28 checking a no is friendly or not (the ratio between the sum of divisors of a number and the number itself i.e ?(n)/n. S o, two number n and m are friendly number if  
?(n)/n = ?(m)/m.  
where ?(n) is the sum of divisors of n)***.*

n1=int(input("Enter first no: "))

n2=int(input("Enter second no: "))

su1,su2=0,0

for i in range(1,n1):

if n1%i==0:

su1=su1+i

for i in range(1,n2):

if n2%i==0:

su2=su2+i

if su1/n1==su2/n2:

print("Amiable friendly")

else:

print("not")

**#29 Anagram together (An anagram of a string is another string that contains same characters, only the order of characters can be different. For example, “abcd” and “dabc” are anagram of each other.)**

def group\_ana(words):

arr=[''.join(sorted(word))for word in words]

print(arr)

dict={}

for i,e in enumerate(arr):

dict.setdefault(e,[]).append(i)

print(dict)

for index in dict.values():

print([words[i]for i in index])

words=[]

n=int(input("Enter the no of words: "))

for i in range(n):

ele=input("Enter a word: ")

words.append(ele)

print("The group anagrams are: ")

group\_ana(words)

**#30 printing Nth ugly Number**

**( Ugly numbers are numbers whose only prime factors are 2, 3 or 5. The sequence 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, … shows the first 11 ugly numbers. By convention, 1 is included)**

n=int(input("Enter Nth ugly no: "))

a=[]

def ugly\_num(n):

a=[0]\*n

a[0]=1

i2=i3=i5=0

mul2=2

mul3=3

mul5=5

for i in range(1,n):

a[i]=min(mul2,mul3,mul5)

if a[i]==mul2:

i2+=1

mul2=a[i2]\*2

if a[i]==mul3:

i3+=1

mul3=a[i3]\*3

if a[i]==mul5:

i5+=1

mul5=a[i5]\*5

return a[-1]

print(ugly\_num(n))

**# 31 program to check if a no is harshad no or not .**

**#(A Harshad number (or a Niven number) is a number that is evenly divisible by the sum of its digits. An example is 18 (1+8=9, 18/9 = 0).)**

n1=int(input("Enter a no: "))

sum=0

t=n1

while n1!=0:

d=n1%10

sum=sum+d

n1=n1//10

if t%sum==0:

print("Harshad no: ")

else:

print("Not a harshad no")

**#32 python program to check if a no is Automorphic no or not:**

**# An Automorphic numbe r is that number whose square ends with the same #digits as the original number. Ex -5, 76, etc.**

n2=int(input("Enter a no to check automorphic no: "))

sq=n2\*n2

d=sq%10

if n2==d:

print("Automorphic")

else:

print("no")

**# 33 Python program to check if a no is abundant no or not(A number is called an abundant number or excessive number if the sum of all of its proper divisors is greater than the number itself.)**

n3=int(input("Enter a no to check no is abundant: "))

sum1=0

for i in range(1,n3):

if n3%i==0:

sum1=sum1+i

if sum1>n3:

print("Abundant no: ")

else:

print("Not")

**# 34 python program to check if two strings are anagram of each other or not**

**An anagram of a string is another string that contains the same characters, only the order of characters can be different.**

s1="Silent"

s2="Listen"

if sorted(s1.lower())==sorted(s2.lower()):

print("Strings are anagram of each other")

**#without sorted function**

count=0

for ch in s1.lower():

for c in s2.lower():

if ch==c:

count+=1

if count==len(s1):

print("strings are anagram ")

else:

print("Not anagram")

**# 35 python program to check if a no is strong no or not**

**Strong numbers are those numbers whose sum of factorial of each digits is equal to the original number. Strong Number Examples: 1 is strong number because 1!=1, 2 is strong number i.e. 2! = 2, 145 is strong number i.e. 1! + 4! + 5! = 1 + 24 + 120 = 145 etc.**

def factorial(n):

fact=1

if n==0 or n==1:

return fact

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

fact=fact\*i

return fact

def find\_strng(lst):

res=[]

for n in lst:

sum=0

temp=n

while(n!=0):

r=n%10

sum+=factorial(r)

n=n//10

if sum==temp:

res.append(temp)

return res

if \_\_name\_\_ == "\_\_main\_\_":

lst=[145,375,100,2,0]

s=find\_strng(lst)

for c in s:

print(c,end=" ")

**# 36 return total no of coins**

n=int(input("Enter day on which you want to calculate charity: "))

def fun(n):

total=0

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

total=total+(i\*\*2)

return total

print(fun(n))

**# 37 Longest Common Sequence problem**

n=int(input("Enter size: "))

a=[]

for i in range(n):

a.append(int(input()))

print(a)

def lis(a):

lis=[1]\*n

for i in range(1,n):

for j in range(0,i):

if a[i]>a[j] and lis[i]<lis[j]+1:

lis[i]=lis[j]+1

maximum=0

for i in range(n):

maximum=max(maximum,lis[i])

return maximum

print(lis(a))

**# 38 finding sum of an array and check if the sum values are in array or not**

def check(\_array):

s=set()

s.add(0)

sum=0

for i in \_array:

sum+=i

if sum in s:

return True

s.add(sum)

return False

if \_\_name\_\_ == "\_\_main\_\_":

size=int(input("Enter size: "))

\_array=list(map(int,input().split(" ")[:size]))

if check(\_array):

print("Yes")

else:

print("No")

**#39 pattern**

s="PUREPYTHON"

for i in range(1,len(s)+1):

print(s[:i])

**# 40 Pattern[Box pattern]**

for i in range(1,5):

for j in range(1,5):

if (i==1 or i==4 or j==1 or j==4):

print("\*",end="")

else:

print(" ",end="")

print()

**#41 Pyramid pattern**

for i in range(1,6):

print(" "\*(5-i)+"\*"\*(2\*i-1))

for i in range(6,1,-1):

print(" "\*(5-i)+"\*"\*(2\*i-1))

**#42 printing swastik pattern**

for i in range(7):

for j in range(7):

if i<7//2:

if j<7//2:

if (j==0):

print("\*",end="")

else:

print(' ',end=" ")

elif j==7//2:

print("\*",end=" ")

else:

if (i==0):

print(" \*",end="")

elif i==7//2:

print("\* ",end="")

else:

if (j==7//2 or j==6):

print("\* ",end="")

elif i==6:

if (j<=7//2 or j==6):

print("\* ",end="")

else:

print(" ",end=" ")

else:

print(" ",end=" ")

print()

**# 43 diamond shape pattern**

for i in range(1,6):

print(" "\*(5-i)+"\*"\*(2\*i-1))

for i in range(6,0,-1):

print(" "\*(5-i)+"\*"\*(2\*i-1))

**# 44 python program for printing window pattern**

for i in range(7):

for j in range(7):

if i==0 or i==6 or j==4:

print("\*",end=" ")

print()

**OUTPUT:**

C:\Users\user\Desktop\python examples>pattern.py

P

PU

PUR

PURE

PUREP

PUREPY

PUREPYT

PUREPYTH

PUREPYTHO

PUREPYTHON

\*\*\*\*

\* \*

\* \*

\*\*\*\*

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\* \* \* \* \*

\* \*

\* \*

\* \* \* \* \* \* \*

\* \*

\* \*

\* \* \* \* \*

\* \*

\* \*

\*\*\*\*\*\*

\* \*

\* \*

\*\*\*\*\*\*

\* \*

\* \*

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

**#45 python program to print all numbers in which each no has has only two factors**

n=int(input("Enter no: "))

a=[]

for i in range(2,n):

count=1

for j in range(1,i):

if i%j==0:

count+=1

if count==2:

a.append(i)

else:

continue

print(a)

**# 46 #7 type of Indian currency notes are given i.e., Rs.1, Rs.2 , Rs.5,Rs.10,Rs.20,Rs.50 and**

**# Rs.100 The function accepts an integer 'n' as its argument .Implement the function to find and return**

**#the minimum no of notes required to form the amount 'n'.**

amount=int(input("Enter the rupees: "))

def currency(amount):

count=0

notes=[100,50,20,10,5,2,1]

notecounter=[0,0,0,0,0,0,0]

for i,j in zip(notes,notecounter):

if amount>=i:

j=amount//i

#print(j)

amount=amount-j\*i

#print(amount,end=" ")

f=i\*j

print(j,"note of Rs."+str(i)+" = "+str(i)+"\*"+str(j)+"="+str(f))

count+=j

print(count)

currency(amount)

####################### 48 ################################################

#FIND maximum and minimum amount to buy all N candies

#In a candy store there are N different type of candies available and the prices of all the N

# different types of candies are provided.There is also an attractive offer by candy store.

# We can buy a single candy from store and get at-most K other candies(all are different types) for free.

#1. Find minimum and maximum amount of money we have to spend to buy all the N different candies.

price=[3,2,1,4]

max=price[0]

for i in range(len(price)):

    for j in range(i+1,len(price)):

        if price[i]>price[j]:

            price[i],price[j]=price[j],price[i]

print(price)

k=int(input("Enter the candy no you want to buy: "))

n=len(price)

index=0

def MIN(price,n,k):

    res=0

    i=0

    while n:

        res+=price[i]

        n=n-k

        i+=1

    return res

def MAX(price,n,k):

    res=0

    index=0

    i=n-1

    while i>=index:

        res+=price[i]

        index+=k

        i-=1

    return res

print(MIN(price,n,k),MAX(price,n,k))

**OUTPUT:**

Enter the rupees: 13

1 note of Rs.10 = 10\*1=10

1 note of Rs.2 = 2\*1=2

1 note of Rs.1 = 1\*1=1

3

[1, 2, 3, 4]

Enter the candy no you want to buy: 2

3 7