Top of Form

Bottom of Form

Top of Form

Bottom of Form

|  |  |
| --- | --- |
|  | ***#1. GCD of two no :*** |
|  | """ |
|  | def gcd(n,m): |
|  | if m==0: |
|  | return n |
|  | else: |
|  | return gcd(m,n%m) |
|  | x=int(input("enter 1st no:")) |
|  | y=int(input("enter 2nd no:")) |
|  | print(gcd(x,y)) |
|  | """ |
|  |  |
|  | ***#2. check prime or not :*** |
|  | """ |
|  | def fprime(n): |
|  | for i in range(2,n): |
|  | if n%i==0: |
|  | print("not prime") |
|  | break |
|  | else: |
|  | print("prime") |
|  | x= int(input("enter a no :")) |
|  | fprime(x) |
|  | """ |
|  | #2. check within a range: |
|  | """ |
|  | def cprime(n,m): |
|  | for n in range(n,m+1): |
|  | if n>1: |
|  | for i in range(2,n): |
|  | if(n%i)==0: |
|  | break |
|  | else: |
|  | print(n) |
|  | x=int(input("enter a no :")) |
|  | y=int(input("enter a no :")) |
|  | cprime(x,y) |
|  | """ |
|  |  |
|  | ***#3.twin prime in a range:*** |
|  | """ |
|  | def prime(n): |
|  | for i in range(2,n): |
|  | if n%i==0: |
|  | return False |
|  | return True |
|  | def twins(start,end): |
|  | for i in range(start,end): |
|  | j=i+2 |
|  | if(prime(i) and prime(j)): |
|  | print("{:d} and {:d}".format(i,j)) |
|  | twins(2,100) |
|  | """ |
|  | #4. nonfibonacci with in a range: |
|  | """ |
|  | import math |
|  | def perfectsquare(n): |
|  | s= int (math.sqrt(n)) |
|  | return s\*s==n |
|  | def fibonacci(n): |
|  | return perfectsquare(5\*n\*n+4) or perfectsquare(5\*n\*n-4) |
|  | for i in range(10,50): |
|  | if(fibonacci(i)==False): |
|  | print(i,"is not a fibonacci no.") |
|  | """ |
|  |  |
|  | ***#5.krishnamurti no:*** |
|  | """ |
|  | def factorial(n): |
|  | fact=1 |
|  | while(n!=0): |
|  | fact=fact\*n |
|  | n=n-1 |
|  | return fact |
|  | def krishnamurti(n): |
|  | sum=0 |
|  | temp=n |
|  | while(temp!=0): |
|  |  |
|  | rem=temp%10 |
|  | sum=sum+factorial(rem) |
|  | temp=temp//10 |
|  | return (sum ==n) |
|  | y=int(input("enter no :")) |
|  | if (krishnamurti(y)): |
|  | print ("yes") |
|  | else: |
|  | print("no") |
|  | """ |
|  |  |
|  | ***#6.prime palindrome no :*** |
|  | """ |
|  | def primepalindrome(a,b): |
|  | for i in range(a,b): |
|  | y=True |
|  | if(str(i)==str(i)[::-1]): |
|  | if(i>2): |
|  | for a in range(2,i): |
|  | if(i%a==0): |
|  | y=False |
|  | break |
|  | if y: |
|  | print(i) |
|  | x=int(input("enter 1st limit")) |
|  | y=int(input("enter 2nd limit")) |
|  | primepalindrome(x,y) |
|  | """ |
|  | ***#7. print 0*** |
|  | ***# 10*** |
|  | ***# 010*** |
|  | ***# 1010*** |
|  | """" |
|  |  |
|  | n = int(input("Please Enter the total Number of Rows : ")) |
|  |  |
|  | def pattern(n): |
|  | for i in range(1, n + 1): |
|  | for j in range(1, i + 1): |
|  | if((i+j)%2==1): |
|  | print('1', end = ' ') |
|  | else: |
|  | print('0', end = ' ') |
|  | print() |
|  | pattern(n) |
|  | """ |
|  |  |
|  | ***#8. print pattern \*\*\*\**** |
|  | ***# \* \*\**** |
|  | ***# \*\* \**** |
|  | ***# \*\*\*\**** |
|  | """ |
|  | def pattern(n): |
|  | for i in range(n): |
|  | for j in range(n): |
|  | if j==0 or j==n-1 or i==0 or i==n-1 or (j==i+1 and j>0) or j==i-1: |
|  |  |
|  | print("\*",end=" ") |
|  | else: |
|  | print(end=" ") |
|  | print() |
|  | m=int(4) |
|  | pattern(m) |
|  | """ |
|  | ***#9. Count no of Non Zero No within a no.*** |
|  | """ |
|  | def count(n): |
|  | count=0 |
|  | while(n > 0): |
|  | r=n%10 |
|  |  |
|  | if(r!=0): |
|  | count=count+1 |
|  |  |
|  | n=n// 10 |
|  | return count |
|  |  |
|  | number=int(input("please enter any number:")) |
|  | count=count(number) |
|  | print("\n Number of digits in a given number =%d" %count) |
|  | """ |
|  | ***#10. Palindrome.*** |
|  | ***"""*** |
|  | def palindrome(a): |
|  | for i in range(a): |
|  | if(str(i)==str(i)[::-1]): |
|  | print("palindrome") |
|  | break |
|  | else: |
|  | print("not palindrome") |
|  | n=int(input("enter no:")) |
|  | palindrome(n) |
|  | """ |
|  |  |
|  | ***#11.perfect no:*** |
|  | ***"""*** |
|  | def perfect(n): |
|  | sum=0 |
|  | for i in range(1,n): |
|  | if(n%i==0): |
|  | sum=sum+i |
|  | if sum==n: |
|  | print("perfect no ") |
|  | else: |
|  | print("not perfect") |
|  | a=int(input("enter number:")) |
|  | perfect(a) |
|  | """ |
|  | ***#12. A power M :*** |
|  | ***"""*** |
|  | def power(base,expo): |
|  | if(expo==1): |
|  | return base |
|  | else: |
|  | return(base\*power(base,expo-1)) |
|  | a=int(input("enter the base value:")) |
|  | b=int(input("enter the exponent:")) |
|  | print("power:",power(a,b)) |
|  | """ |
|  | ***#13. Factor of a no:*** |
|  | ***"""*** |
|  | def factor(n): |
|  | num=1 |
|  | while(num<=n): |
|  | if(n%num==0): |
|  | print(num) |
|  | num=num+1 |
|  | n=int(input("enter no:")) |
|  | factor(n) |
|  | """ |
|  | ***#14. armstrong:*** |
|  | ***"""*** |
|  | def armstrong(n): |
|  | sum=0 |
|  | m = n |
|  | while(m>0): |
|  | r=m%10 |
|  | sum=sum+r\*\*3 |
|  | m //= 10 |
|  | if(n == sum): |
|  | print("ARMSTRONG") |
|  | else: |
|  | print("Not ARMSTRONG") |
|  | n=int(input("enter no:")) |
|  | armstrong(n) |
|  | """ |
|  |  |
|  | ***#15. Fibonacci:*** |
|  | ***"""*** |
|  | def fib(n): |
|  | if n<=1: |
|  | return n |
|  | else: |
|  | return(fib(n-1)+fib(n-2)) |
|  | m=int(input("enter how many terms:")) |
|  | for i in range(m): |
|  | print(fib(i)) |
|  | """ |
|  | ***#16. Lucas No:*** |
|  | ***"""*** |
|  |  |
|  | def lucas(n) : |
|  | if (n==0): |
|  | return 2 |
|  | if (n==1): |
|  | return 1 |
|  | return lucas(n - 1) + lucas(n - 2) |
|  | m=int(input("enter terms:")) |
|  | print(lucas(m)) |
|  | """ |
|  |  |
|  |  |
|  |  |
|  |  |