1. r=float(input("Enter the radius : "))

if 0<=r<=100:print("Area of circle =",f"{3.14*r*r:.6f}")

else:print("Enter a positive value upto 100")

2. L=int(input("L:"))

B=int(input("B:"))

Make use of the values of L and B read using the input function.

print(L*B)

S=int(input("S:")) # Make use of the value of S read using the input function.

print(S*S)

1. base = float(input("Base: "))

height = float(input("Height: "))

area = 0.5 * base * height

print(f"Area: {area:.2f}")

2. import math

a=int(input('a: '))

b=int(input('b: '))

c=int(input('c: '))

d=b**2-4*a*c

if d>0:

r1=(-b + math.sqrt(d))/(2*a)

r2=(-b - math.sqrt(d))/(2*a)

print((f"The roots are: {r1:.2f} and {r2:.2f}"))

```
elif d==0:
root=-b/(2*a)
print(f"The root is: {root:.2f}")
else:
real=b/(2*a)
img = math.sqrt(abs(d))/(2*a)
print(f"The roots are: {-real:.2f}+{img:.2f}j and {-real:.2f}-{img:.2f}j")
```

3. # Type Content here...

```
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))
largest = max(num1, num2, num3)
print("Largest number:",largest)
```

4. C = float(input("celsius: "))

```
F = 1.8 * C + 32
print("fahrenheit:", F)
```

5. a = list(map(int,input("Set A: ").split()))

```
A = set(a)
b = list(map(int,input("Set B: ").split()))
B = set(b)

# Write your code here to perform different operations
print("Union: ", A | B)
print("Intersection: ", A & B)
print("Difference: ", A - B)
```

6. year = int(input('Enter a year: '))

```
if(year%4==0):
print(year, 'is a leap year')
else:
print(year, 'is not a leap year')
```

7. s1=float(input('subject 1: '))

```
s2=float(input('subject 2: '))
s3=float(input('subject 3: '))
s4=float(input('subject 4: '))
s5=float(input('subject 5: '))
avg = (s1+s2+s3+s4+s5)/5
print('Average Marks:',format(avg,'.2f'))
if (avg>=90 and avg<=100):
print('Grade: A')
elif(avg>=80 and avg<=100):
print('Grade: B')
elif(avg>=70 and avg<=79):
print('Grade: C')
elif(avg>=60 and avg<=69):
print('Grade: D')
elif(avg<60):
print('Grade: F')
```

8. year= int (input ('year: '))

```
m= int(input('month: '))
d=int(input('day: '))
leap =(year % 4==0 and year % 100 !=0)or (year %400 == 0)
month_days= [31,28+leap,31,30,31,30,31,31,30,31,30,31]
```

```
if m<1 or m>12 or d<1 or d>month_days[m-1]:
print("invalid")
else:
print("valid")
d+=1
if d>=month_days[m-1]:
d=1
m+=1
if m>12:
m=1
year+=1
print(f"incremented date: {year}-{m:02d}-{d:02d}")
```

9.
```
n=int(input ("Enter a number : "))
if n<0:
print("Enter a positive number")
else:
fact=1
for i in range (1,n+1):
fact *=i
print(f"Factorial of given number is : {fact}")
```

10.
```
num = int(input("Enter a number : "))
for i in range(num,0,-1):
print("* "* i)
```

11.
```
digit1=int(input("digit1 (0-9): "))
digit2=int(input("digit2 (0-9): "))
```

```
digit3=int(input("digit3 (0-9): "))
if(0<=digit1<=9 and 0<=digit2<=9 and 0<=digit3<=9):
digits=[digit1,digit2,digit3]
print(f"{digits[0]}{digits[1]}{digits[2]}")
print(f"{digits[0]}{digits[2]}{digits[1]}")
print(f"{digits[1]}{digits[0]}{digits[2]}")
print(f"{digits[1]}{digits[2]}{digits[0]}")
print(f"{digits[2]}{digits[0]}{digits[1]}")
print(f"{digits[2]}{digits[1]}{digits[0]}")
else:
print("Invalid")
```

12.
```
def matmult(A, B):
rows_A = len(A)
cols_A = len(A[0])

rows_B = len(B)
cols_B = len(B[0])

if cols_A != rows_B :
print("Cannot multiply the two matrices. Incorrect dimensions.")
return None

result = []
for i in range(rows_A):
row=[]
for j in range(cols_B):
row.append(0)
result.append(row)
```

```
for i in range(rows_A):
for j in range(cols_B):
for k in range(cols_A):
result[i][j]+=A[i][k]*B[k][j]
return result

def readmatrix(name=''):
print(f"Enter values for {name}")
rows=int(input("Number of rows, m = "))
cols=int(input("Number of columns, n = "))

matrix = []

for i in range(rows):
row=[]
for j in range(cols):
print(f"Entry in row: {i+1} column: {j+1}")
value= int(input())
print("Matrix - A * Matrix- B =",matmult(matrixa,matrixb))
```

13.
```
is_ def prime(num):
if num<2:
return False
for i in range(2,num):
if num%i==0:
return False
return True
upper_limit=int(input("Enter upper limit: "))
for i in range(2,upper_limit):
if is_prime(i):
```

```
print(i)
```

14.
```
def vowel_count(str):
count = 0
vowel = set("aeiouAEIOU")
for char in str:
if char in vowel:
count+=1
return count
# Write your code here to count the vowels
str = input()
vowel_count(str)
print(vowel_count(str))
```

15.
```
string=input()
if string==string[::-1]:
print("palindrome")
else:
print("not a palindrome")
```

16.
```
s =input("")
m_s=""
for char in s:
if char.isalnum() or char.isspace():
m_s+=char
print(m_s)
```

17. string=input()

```
r=string[::-1]
print(r)
```

18. def sum(num):

```
sum=0
for i in num:
sum+=int(i)
print("sum:",sum)
num=input("num: ")
sum(num)
```

19. def Sumof(n):

```
if n==0:
return 0
else:
return n%10+Sumof( n// 10 )
# take user input and add the function call
n=int(input())
result=Sumof(n)
print(result)
```

20. '''while True:

```
print("1. Add Contact")
print("2. Remove Contact")
print("3. Display")
print("4. Quit")
'''
```

```
phonebook = {}

while True:
print("1. Add Contact")
print("2. Remove Contact")
print("3. Display")
print("4. Quit")

choice = int(input("Enter choice (1-4): "))

if choice == 1:
name = input("Name: ")
phone_number = input("Phone number: ")
phonebook[name] = phone_number
elif choice == 2:
name = input("Name: ")
if name in phonebook:
del phonebook[name]
#(f"Contact '{name}' removed.")
else:
print("Not found")
elif choice == 3:
print(phonebook)
elif choice == 4:
break
else:
```

21. import fibonacci_module

```
def main():
try:
n = int(input("Enter the max value: "))

if n > 0:
fibonacci_series = fibonacci_module.generate_fibonacci_sequence(n)
print(f"Fibonacci series upto {n} :")
print(*fibonacci_series,end=(" "))
else:
print("Please enter a positive integer")

except ValueError:
print("Invalid input. Please enter an integer.")

if __name__ == "__main__":
main()
```

22.
```
class Complex ():
def initComplex(self,num):
print(f"complex number {num}")
print('Real Part:',end=" ")
self.real=int(input())
print('Imaginary Part:',end=" ")
self.imag = int(input())
def display(self):
print(f"{self.real}+{self.imag}i")
def sum(self,c1,c2):
self.real= c1.real+c2.real
self.imag = c1.imag + c2.imag
```

```
c1 = Complex()
c2 = Complex()
c3 = Complex()
c1.initComplex(1)
c1.display()
c2.initComplex(2)
c2.display()
print('Sum:',end=' ')


c3.sum(c1,c2)
c3.display()
```

26. class Car:

```
def __init__(self,brand,price,model,color):
self.brand =brand
self.model = model
self.price=price
self.color=color
def display_details(self):
print(f"{self.brand}")
print(f"{self.price}")
print(f"{self.model}")
print(f"{self.color}")

class Car1(Car):
def diaplay_details(self):
```

```
super().display_details()
class Car2(Car):
def display_details(self):
super().display_details()




def get_car_details():
brand = input("brand: ")
price = float(input("price: "))
model = input("model: ")
color = input("color: ")
return brand, price, model, color

print("Details for Car 1:")
car1_brand, car1_price, car1_model, car1_color = get_car_details()

# create an object car1
```