1. write a program in ruby to multiply all no.s in list

```
A = [1, 2, 3, 4, 5]
result = 1
A.each do |i|
   if i!= 0
      result = result*i
   else
      result
   end
end
puts result
```

2.write a program in ruby to find square of any no. using function

```
i=1
for i in(1..5)
   puts(i*i*i)
   i=i+1
end
```

3.write a python prgm to check given integer is positive even/odd,negative even/odd,print even if integer is zero using if else ladder

```
num = 10
if num > 0:
print("Positive number")
elif num == 0:
print("Zero")
else:
print("Negative number")

# Checking for odd and even
if (num % 2) == 0:
print("{0} is Even".format(num))
else:
print("{0} is Odd".format(num))
```

4.python prgm to take value from user as input any character and check whether it is alphabet,digit or special character using case when stmt

```
ch=input("Please enter your own chracter")
```

```
if((ch>='a' and ch<='z')or(ch>='A' and ch<='Z')):
    print("given chracter",ch, "is an alphabet")
elif(ch>='0' and ch<='9'):
    print("given chracter",ch,"is a digit")
else:
    print("given chracter",ch,"is a special chracter")
```

5.write a program in ruby to find cube of any no. up to given no.

```
i=1
for i in(1..5)
    puts(i*i*i)
    i=i+1
end
```

6.using switch write a python prgm to input marks of five sub and calculate percentage and grade

```
phy=int(input("Enter the Marks in Physics:\n"))
chem=int(input("Enter the Marks in Chemistry:\n"))
bio=int(input("Enter the Marks in Biology:\n"))
math=int(input("Enter the Marks in Mathematics:\n"))
comp=int(input("Enter the Marks in Computer:\n"))
per=(phy+chem+bio+math+comp)*100/500
print(per)
if(per>=90):
    print("Grade A")
elif(90>per>=80):
    print("Grade B")
elif(80>per>=70):
    print("Grade C")
elif(70>per>=60):
    print("Grade D")
elif(60>per>=40):
    print("Grade E")
else:
    print("Fail!!")
```

7.write a python program to find sum of first 10 natural no.

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

if num < 0:
   print("Enter a positive number")
else:
   sum = 0
   # use while loop to iterate un till zero
   while(num > 0):
      sum += num
      num -= 1
   print("The sum is",sum)
```

8.implement simple dml n ddl command using workbench.create database and tables execute simple queries

```
import mysql.connector
mydb=mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="student_info"
    )

mycursor=mydb.cursor()
mycursor.execute("drop table if exists students")
print("Table dropped")

mycursor.execute("create table students(id int auto_increment primary key, name varchar(25), age varchar(5))")
print("table created")

sql="insert into students(name,age) values(%s,%s)"
val=[
            ('Rakesh','20'),
        ('Shradha','21'),
        ('Vaishnavi','22'),
        ('Jui','19')


]
mycursor.executemany(sql,val)
mydb.commit()
```

```
print("Record inserted")
```

```
import mysql.connector
conn=mysql.connector.connect(
 host="localhost",
 user="root",
 password="")
print(conn)
curr=conn.cursor()
curr.execute("create database customer")
print("database created.....!")

import mysql.connector
conn = mysql.connector.connect(user='root',password='',host='localhost')
cursor = conn.cursor()
try:
 #cursor.execute("show tables")
 cursor.execute("use customer")
 sql="insert into
customer(Name,Id,Phone_No,Address)values(%s,%s,%s,%s)"

val=[('gayatri','3001','8521128952','peth vadgaon'),('samiksha','3002','9960648153','Wardha')]
cursor.executemany(sql,val)
 print("Data inserted successfully")
 """for i in cursor:
 print(i)"""
 conn.commit()
except:
 conn.rollback()
finally:
 print("Connection Successful")
 print(conn)
```

9.write a python program to display no. in reverse order without using function

```
num = 123456
print(str(num)[::-1])
```

10.python program to take value from user as input basic salary of employee and calculate

gross salary

```
bs = float(input(“Enter the Basic Salary :”))
if(bs<1500):
        Hra=bs*0.1
        Da=bs*0.9
else:
        Hra=500
        Da=bs*0.98

gs=bs + Hra + Da
print(“Gross Salary Rs :”,gs)
```

11.python prgm to create calculator using case when

```
def add(x, y):
   return x + y

# This function subtracts two numbers
def subtract(x, y):
   return x - y

# This function multiplies two numbers
def multiply(x, y):
   return x * y

# This function divides two numbers
def divide(x, y):
   return x / y


print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

while True:
   # take input from the user
   choice = input("Enter choice(1/2/3/4): ")

   # check if choice is one of the four options
```

```
    if choice in ('1', '2', '3', '4'):
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))

        if choice == '1':
            print(num1, "+", num2, "=", add(num1, num2))

        elif choice == '2':
            print(num1, "-", num2, "=", subtract(num1, num2))

        elif choice == '3':
            print(num1, "*", num2, "=", multiply(num1, num2))

        elif choice == '4':
            print(num1, "/", num2, "=", divide(num1, num2))

        # check if user wants another calculation
        # break the while loop if answer is no
        next_calculation = input("Let's do next calculation? (yes/no): ")
        if next_calculation == "no":
          break

    else:
        print("Invalid Input")
```

12.Write a Python function student data () which will print the id of a student (student_id). If the user passes an argument student_name or student_class the function will print the student name and class

```
def student(student_id, student_name, student_class):
    return f'Student ID: {student_id}\nStudent Name: {student_name}\nClass: {student_class}'
print(student('3014', ‘Gayatri Jadhav’, 'TY Btech'))
```

13.python prgm to take value from user as input electricity unit charges and calculate according to given condition for first 50 units rs 0.50/unit

```
def electricitybill():
    if(100>unit):
        amount=unit*2.5
```

```
        charge=20
    elif(300>unit>=100):
        amount=unit*3.5
        charge=50
    elif(500>unit>=300):
        amount=unit*5.5
        charge=70
    elif(700>unit>=500):
        amount=unit*7.5
        charge=100
    elif(unit>=700):
        amount=unit*10.5
        charge=150
    print("Electricity Bill =",amount+charge,"Rs")

unit=int(input("Enter the unit:"))
electricitybill()
```

14. Write python program to perform operations on lists, dictionaries, tuple,sets

```
#List
List=[1,2,'third',5,8,'box']
print(List)
print(List[1])
print(List[-2])
print(len(List))
List.append(6)
List.append(8)
print(List)
List.insert(4,"paper")
print(List)
List.reverse()
print(List)
List.remove(2)
print(List)
List.pop()
print(List)
sliced_list=List[4:]
print(sliced_list)
print("--------------------------------------------------")
#Tuple
tuple1=(1,2,3,4,5,5,"pen")
```

```
tuple2=(7,78,4)
print(tuple1)
#print(cmp(tuple1,tuple2))
print(len(tuple1))
print(max(tuple2))
print(min(tuple2))
print(tuple1.count(5))
print(tuple1)

print("-------------------------------------------------")
#Dictionary
dict1={ 1:"one",2:"two",3:"three",4:"four"}
print(dict1.get(1))
print(dict1.keys())
print(dict1.values())
print(dict1.pop(2))
print(dict1)
print("-------------------------------------------------")
#sets
sets1={1,2,3,4,5,7}
sets2={6,7,8,9,10,4,2}
print(sets1)
print(sets2)
print(sets1.union(sets2))
print(sets1.intersection(sets2))
print(sets1.difference(sets2))
sets1.add(11)
print(sets1)
print(sets1.isdisjoint(sets2))
sets1.clear()
print(sets1)
print("-------------------------------------------------")
```

15.Create a child class bus that will inherit all of the variable and methods of the vehicle class

```
class Vehicle():
    def description(self):
        print("i am vehicle")
class Car(Vehicle):
    def description(self):
        print("i am car")
class Bus():
```

```
    def description(self):
        print("i am Bus")
class Truck():
    def description(self):
        print("i am Truck")

v=Vehicle()
c=Car()
b=Bus()
t=Truck()

v.description()
c.description()
b.description()
t.description()
```

16.Print right angle triangle pattern of asterisk

```
def pattern(n):
    for i in range(0,n):
        for j in range(0,i+1):
            print(" * ",end=" ")
        print("\r")
n=5
pattern(n)
```

17.Write a ruby program to find out given year is leap year or not

```
puts "Enter the year you want to check"
yr = gets.chomp.to_i

if yr % 400 == 0
        puts "#{yr} is a leap year"
elsif yr % 4 == 0 && yr % 100 !=0
        puts "#{yr} is a leap year"
else
        puts "#{yr} is not a leap year"
end
```

18.write ruby program to display even no. from 1 to 50

```
puts "Even numbers between 1 to 50: "
```

2.step 50, 2 do |x|
    puts "#{x}"
 end

19.write ruby program to find greatest no. from given 3 no.

```
x,y,z = 2,5,4
if x >= y and x >= z
    puts "x = #{x} is greatest."
elsif y >= z and y >= x
    puts "y = #{y} is greatest."
else
    puts "z = #{z} is greatest."
end
```

20.write python program to maximum between 3 no.s using if elsif ladder.

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

if (num1 >= num2) and (num1 >= num3):
   largest = num1
elif (num2 >= num1) and (num2 >= num3):
   largest = num2
else:
   largest = num3

print("The largest number is", largest)
```

21.write python prgm to calculate bikes average consumption from given total distance travelled in km and spent fuel

22.ruby employee

```
#Employee payroll Details
module Employee
   def details
     puts "Enter Employee Id"
     @emp_id=gets.to_i
     puts "Enter Employee Name"
     @emp_name=gets
```

```
      puts "Enter Date of Birth"
      date=gets.to_i
      puts "Ente the Employee Salary"
      @emp_salary=gets.to_i
    end
end
module Salary
    def salaries
      @da = @emp_salary*15/100
      @hra = @emp_salary*10/100
      @pf = @emp_salary*12/100
      @esi = @emp_salary*2/100
      @deduction=@pf + @esi
      @gross_sal=@emp_salary + @da + @hra
      @net_sal=@gross_sal- @deduction
        puts "\nEmployee Id = #{@emp_id}"
        puts "Employee Name = #{@emp_name}"
        puts "\nGross Salary = $#{@gross_sal}"
        puts "Employee DA  = $#{@da}"
        puts "Employee HRA  = $#{@hra}"
        puts "Employee PF deduction = $#{@pf}"
        puts "Employee ESI deduction = $#{@esi}"
    end
end
module Hours
  def pay_per_hour
    pay_for_period=(@emp_salary / 365) * 12
      puts "\nPay per hour = $#{pay_for_period}"
      puts "\nYour Net Salary = $#{@net_sal}"
  end
end
class Pay
  include Employee
  include Salary
  include Hours
  def role
    if @emp_salary == 0
      puts "A salary of $#{@emp_salary} isn't valid!"
    else
      puts "\nPlease type Your Bank A/C"
      bank=gets.to_i
      puts "\nYour Net Salary = $#{@net_sal} is Credited in Your account"
      puts "\nHave a Nice Day!!"
```

```
    end
  end
end
obj1=Pay.new
obj1.details
obj1.salaries
obj1.pay_per_hour
obj1.role
```

23.palindrome

```
num=1221
temp=num
reverse=0
while temp>0:
    remainder=temp%10
    reverse=(reverse*10)+remainder
    temp=temp//10
if num==reverse:
    print('Pallindrome')
else:
    print('Not Pallindrome')
```

24.operators in python

```
A = 7
B = 2

# addition
Print (‘Sum: ‘, a + b)

# subtraction
Print (‘Subtraction: ‘, a – b)

# multiplication
Print (‘Multiplication: ‘, a * b)

# division
Print (‘Division: ‘, a / b)

# modulo
Print (‘Modulo: ‘, a % b)
```

```
# a to the power b
Print (‘Power: ‘, a ** b)
```

25.python prgm prime no. or not

```
def prime(num):
    n=0
    for i in range(2,n):
        if(num%i==0):
            n=1
            break
    if n==0:
        print("It is prime number")
    else:
        print("It is not a prime number")
prime(2)
```

26.reverse of string

```
string=("Gayatri")
str=""
for i in string:
    str=i+str
print("reverse String",str)
```

27.mail sending

```
from smtplib import*
import smtplib,ssl
obj=smtplib.SMTP_SSL("smtp.gmail.com",465)
obj.login("gayatrisjadhav07@gmail.com","adrexcywftlmzeuc")
obj.sendmail("gayatrisjadhav07@gmail.com","sayalimane8698@gmail.com","Hiii nalayk tuz ky
zalay sgla amche vande ahet")
obj.quit()
```

28.write ruby program that reads 5 numbersand counts the positive number and find average of that positive number