Q Count number of occurances

Prime number

sum of n even number

inverted pattern

Binary to decimal

leap year

Q, Procedure

```
declare two variables a and b
  set a to non-zero value
  set b to 0
  try block:
    attempt to divide a by b and store the result
in another variable result
exception:
  Catch the Arithmetic exception
  display an error message
  end try. catch block
```

Q, Array Bound of exception

Procedure

```
Initialize arr [] = {5, 6, 7, 8, 9}
Set a variable index to a value greater than
the array's length
try block:
```

```
Attempt to access the array element at index
exception

   catch ArrayIndexOutOf Bound Exception
   display error message
   end try catch block
   end Procedure
Q Procedure
define
        Negative Values Exception extending exception
Constructor:
        Send message to parent class
main:
   define check Positive (int number)
      if number < 0
         throw Negative Value Exception
   try:
      call check Positive (number)
   if no exception is thrown "Number is valid"
   catch Negative value Exception:
      display the error
Q declare
      int a=3
      float b=4.56
      double c=5.6473467
```

```
        char d : 'a'
        bool e   true
    display the data types  Seperately
```

Q) Procedure.

```
Initialize   variables
  key  :  make  : model year
Constructor
    input: make , model , year
    Set class  Variable  to input  variable
  Method  PrintCar Details
Output: display  make  model and year of the
        Car

main:
     Create  an  object of  Car with  make. model
and year
     call  printDetails  and  display the details
```

Q

```
Create  a  matrix  A with  dimension  1x9
Create  Matrix  B  with  dimension 1x9
   for  i   in  range  from  0 to 8
           Set  matrix   A [0] [i]  to  i+1
       for  i   in  range  from  9 to 1
```

```
        [illegible] matrix C[a][i] to i
for  i  from  a to 8
        Print  matrix  A[a][i]
    for  i  from  9 b 1
        Print  matrix  B[a][i]
```

Q)

```
Procedure
Initialize    accountnumber , balance
Constructor
    input   account number , initial balance set
account number and balance to input value
Method deposit:
    input amount
    if amount 2= balance
        balance =- amount
    else
    display insufficent balance
main:
    Create object of Bank Account with account
number and initial Balance
    display the balance
```

```
Q) declare the Scanner class to get input
   get the operator and operand
   if operator : Sum
       display the sum
   else if operator == difference
       return the difference between the numbers
   else if operator == +
       return the product
   else if operator == \
       return the quotient
   display the result.
```

```
Q) Procedure

   declare Scanner class
   initialize base = 4
              exponent = 3

     using the intait function
     assign pow = Math.pow (base, exponent)
     display the result

     end procedure
```

Q

```
Procedure:
assign n = length
expected sum = $n/(n+1)/2$
actual - sum = sum (numbers)
missing number = expected - sum - actual - sum
return missing number
end Procedure
```