D. Arrays/iterator and reducers Exercise

A.Write a Ruby program for each of the following to get it out of an array of integers.

ex:[1,2,3,4,5]

a.Max #=> 5

```
arr = [1,2,3,4,5]
p arr.max()
```

b.Min #=> 1

```
arr = [1,2,3,4,5]
p arr.min()
```

c.Count #=>5

```
arr = [1,2,3,4,5]
 p arr.count()
```

d.Sum #=> 15

```
arr = [1,2,3,4,5]
p arr.sum()
```

e.Multiplication of all elements #=> 120

```
arr = [1,2,3,4,5]
multi = 1
arr.each {|element| multi = element * multi}
p multi
```

B.Write a Ruby program to check whether a value exists in an array using array APIs.

```
a = ['a', 'b', 2, 'x']
p a.include?('x')
```

C.Write a ruby program that gets the count of each element in the array.ex: if
a=[1,1,2,2,3,3,3] the output will be result=[[1,2],[2,2],[3,3]]

```
arr1=[1,1,2,2,3,3,3]
arr2 = []
ta = arr1.tally

ta.each {|element| arr2.push(element)}

p arr2
```

D.Write two Ruby programs to compute the sum of elements in a given array of integers using iterators/ reducers

```
arr=[1,2,3]
sum = 0
arr.each {|element| sum = element + sum}

p "Sum by iterator: "+sum.to_s

p "Sum by reducer: "+(arr.reduce(0) { |sum, n| sum + n } ).to_s
```

E.Write a Ruby program to remove duplicate elements from a given array using array
APIs.

```
arr = [1,1,2,2,3,3]
u = arr.uniq()
p u
```

F.Write a Ruby program to create a new array of length 2 containing the middle two elements from a given array of integers of even length 2 or more using arrays APIs

```
arr1 = [1,2,3,4,5,6,7,8]

arr2 = [arr1[(arr1.length/2)-1],arr1[arr1.length/2]]

p arr2
```

G.Write a Ruby program to find the largest value from a given array of integers of odd length. The array length will be a least 1 using array APIs.

```
arr1 = [1,2,3,4,5,6,7]
if arr1.length.odd?
 p arr1.max
 else
 p "not odd length"
 end
```

H.Write a Ruby program to compute the sum of the numbers of a given array except the
number 17 and numbers that come immediately after a 17. Return 0 for an empty array
using array APIs.

```
arr = [1,2,3,4,5,6,7,17]
sum = 0
arr.each { |element|
 if element != 17
    sum = sum + element
 else
    break
 end
 }

p sum
```

I.Write a Ruby program to check whether all items are identical in a given array using
array APIs.

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

  if arr.count(arr[0]) != arr.length
    p "not identical"
else
    p "identical
end
```

J.Given an input text output it transposed.
Roughly explained, the transpose of a matrix:
ABC
DEF
is given by:
AD
BE
CF

```
   p "Enter a word"
input1 = gets.chomp
p "Enter a secound word with the same length"
input2 = gets.chomp

arr1 = [input1.split(""), input2.split("")]

arr2 = arr1.transpose

arr2.each {|e| p e[0].to_s+e[1].to_s}
```