E.Hashes/iterators and reducers Exercise

A.For the following hash:

```
Employees = {
10:{name:”Ahmed”,salary:1000},
21:{name:”Mohamed”,salary:2000},
113:{name:”Mahmoud”,salary:5000},
4:{name:”Yassin”,salary:3000},
52:{name:”Taha”,salary:4000},
102:{name:”Khadija”,salary:nil},
64:{name:”Sara”,salary:5000},
37:{name:”Nadine”,salary:5000},
88:{name:”Doaa”,salary:4000},
90:{name:”Iman”,salary:4000},
103:{name:”Khadija”,salary:1000},
12:{name:”Kholod”,salary:nil},

15:{name:”Said”,salary:nil},
38:{name:”Nadine”,salary:5000},
89:{name:”Doaa”,salary:4000},
91:{name:”Iman”,salary:4000},
104:{name:”Khadija”,salary:1000},
17:{name:”Kholod”,salary:nil},
14:{name:”Said”,salary:nil},
}
```

Write a program for each of the following that

I.Gets all employees name

```
   employees.each {|element| p element[1][:name]}
```

II.Gets all employee IDs ex:[10,21,113,…]

```
   employees.each_key do |key|
  p key
end
```

III.Gets employees with the highest salary in an array along their ID

ex::[{name:”Mahmoud”,salary:5000,id:113},{name:” Sara”,salary:5000,id:64},…]

```
   employees1 = Hash.new()
    employees.each {lelementl
    if element[1][:salary] != nil
        employees1[element[0]] = element[1]
    end
   }
    p employees1.max_by(2) {lk,vl v[:salary]}
```

IV.Gets employees with lowest salary in a hash keeping their IDs

Ex:{
10:{name:”Ahmed”,salary:1000},
103:{name:”Khadija”,salary:1000},
....
}

```
     employees1 = Hash.new()
     employees.each {lelementl
      if element[1][:salary] != nil
        employees1[element[0]] = element[1]
    end
    }
p employees1.min_by(2) {lk,vl v[:salary]}
```

V.Gets average salaries

```
   employees1 = Hash.new()
     employees.each {lelementl
      if element[1][:salary] != nil
        employees1[element[0]] = element[1]
    end
   }
p employees1.sum {lk,vl v[:salary]} / employees.length
```

VI.Remove employees with nil salary.

```
      employees1 = Hash.new()
     employees.each {lelementl
      if element[1][:salary] != nil
        employees1[element[0]] = element[1]
    end
    }
    p employees1
```

VII.Gets hash a new hash with uniq employees (remove duplicate)

```
    p employees.uniq {lel e[1][:name]}
```

B.Write a Ruby program to find most occurred item in a given array along the elements
frequency:
ex:Original array:
[10, 20, 30, 40, 10, 10, 20]
Frequency of numbers:
{10=>3, 20=>2, 30=>1, 40=>1}

```
h = [10, 20, 30, 40, 10, 10, 20]
p h.tally
```