Question 1

Input:

```
{
    "Pick": "4048",
    "Line": 1,
    "Quantity": 1
}
```

Output:

```
[
  {
    "Pick" : "4048"
  },
  {
    "Line" : 1
  },
  {
    "Quantity": 1
  }
]
```

Solution :

```
%dw 2.0
output application/json
var data = {
    "Pick": "4048",
    "Line": 1,
    "Quantity": 1
}
---
```

data pluck ((value, key, index) -> {(key):value})

Question 2

Input: ["United States","Saudi","Australia"]

Output : USA

Solution : payload map($[0]) joinBy ""

Question 3

Input:-

[

{

"id":"101",

"name":"Raghav"

},

{

"id":"102",

"name":"sandeep"

}

]

Output:-

{

"id1": "101",

"name1": "shubham",

"id2": "102",

"name2": "mule"

}

Solution:

(payload reduce ((item, accumulator={}) -> accumulator++ item)) mapObject ((value, key, index) -> {(key++ (index+1)) : value})

Question 4

Input : ["a","a","b","b","c","b"]

Output : ["a2","b3","c1"]

Solution :

(payload groupBy ((item, index) -> item))pluck ((value, key, index) -> key++(sizeOf(value)))

Default value syntax:

```
{
  "userId": payload.id default "0000",
  "userName": payload.name default "Undefined"
}
```

Question 5:

Input : [1,2,3,4,5,6] **Output**: 21 (Sum of all the elements in the array of integers)

Solution :

payload reduce ((item, accumulator=0) -> item + accumulator)

Question 6 :Even Odd question

Input : [1,2,3,4,5,6]

Output: { Even : [2,4,6] , Odd : [1,3,5] }

Solution :

```
{
   "Even" : payload filter ((item, index) -> (item mod 2)== 0),
   "Odd" : payload filter ((item, index) -> (item mod 2)!= 0)
}
```

Question 6 : Update the name question

Input :

```
{
   "name" : "Nity",
   "age" : 25
}
```

Output:

```
{
  "name": "Ganesh",
  "age": 25
}
```

Solution:

```
payload  update {
   case .name -> "Ganesh"
}
```

Question 7:

Input :

```
{
   "name" : "Nity",
   "age" : 25,
```

```
  "address":{
     "Street" : "1st Street",
     "zipCode":"ABCD"
  }
}
```

Output :

```
{
 "name": "Ganesh",
 "age": 25,
 "address": {
  "Street": "2nd Street",
  "zipCode": "ABCD"
 }
}
```

Solution :

```
payload update {
  case .name -> "Ganesh"
  case .address.Street -> "2nd Street"
}
```

Question 8 : "Hello World" to "World Hello"

Solution : ((payload splitBy " ")[-1 to 0])joinBy " "

Question 9:

Input:

```
{"input1" : [{
```

```
   "id" : 1,
   "name" : "Alice"
},
{
   "id" : 2,
   "name" : "John"
},
{
   "id" : 3,
   "name" : "Frank"
}],
"input2" : [
   {
      "id" : 1,
      "age" : 25
   },
   {
      "id" : 2,
      "age" : 30
   }
]}
```

Ouput :

```
[
 {
   "id": 1,
   "name": "Alice",
   "age": 25
 },
```

```
 {
   "id": 2,
   "name": "John",
   "age": 30
 },
 {
   "id": 3,
   "name": "Frank",
   "age": "No age mentioned"
 }
]
```

Solution :

```
(payload.input1 map ((item, index) -> {"id":item.id,"name":item.name,"age":
((payload.input2 filter ((item2, index2) -> item2.id == item.id))[0].age)  default "No age
mentioned"}))
```

Question 10:

Input:

```
[{
   "name" : "John",
   "city" : "LA",
   "company" : "Jio"
},
{
   "name" : "John",
   "city" : "LA",
   "company" : "TCS"
```

}]

Output :

```
{
  "name": "John",
  "city": "LA",
  "company": [
    "Jio",
    "TCS"
  ]
}
```

Solution :

```
{
"name": payload[0].name,
"city": payload[0].city,
"company": payload map ((item, index) -> item.company reduce ((item, accumulator="") -> accumulator ++ item))
}
```

Question 11:

Input :

```
[
   {
    "Name": "Raju",
    "Age" : "25",
    "employeeCode" : "1234"
    }
]
```

Output :


```
[
  {
  "Key" : "Name",
  "Value" : "Raju"
  },
  {
  "Key" : "Age",
  "Value" : "25"
  },
  {
  "Key" : "EmployeeCode",
  "Value" : "1234"
  }
]
```


Solution:

```
payload map ((item, index) -> item pluck ((value, key, index) -
>{"Key":key,"Value":value} )) reduce ((item, accumulator) -> accumulator ++ item )
```