Subject: IaC

In this test, consider that the input will a json data file representing users in this format:

```
{
 "users": [
   {"name": "john", "age":  23, "city": "barcelona"},
   {"name": "bob", "age":  29, "city": "london"}
 ]
}
```

1.Your **first assignment** will be to generate a password for all the users within the json input. Next, you will have to output this as yaml encoded data in the output `test1` in the following format:

```
users:
   - "key": "user[0].name"
     "value": "john"
   - "key": "user[0].age"
     "value": "23"
   - "key": "user[0].city"
     "value": "barcelona"
   - "key": "user[0].password"
     "value": "5]L+J7rA*<7+:PO6"
   - "key": "user[1].name"
     "value": "bob"
   - "key": "user[1].age"
     "value": "29"
   - "key": "user[1].city"
     "value": "london"
   - "key": "user[1].password"
     "value": "P=x&385YGMI0?!Is"
```

The goal behind is to be able to reuse this **data structure** as a data source for another component, and to expose a new output `test2`, that output is a json data structure of users by city, sorted by age

```
{
 "barcelona": [
   {"name": "john", "age":  23, "password": "5]L+J7rA*<7+:PO6"}
```

```
 ],
 "london": [
   {"name": "bob", "age":  29, "password": "P=x&385YGMI0?!Is"}
 ]
}
```

2. Your **second assignment** will be to generate a markdown file containing the users in a table as follows:

```
| Name | Age  | City      | Password safe |
| ---- |:----:| :---------|:--------------|
| John |  23  | Barcelona |true           |
| Bob  |  29  | London    |true           |
```

A password is considered safe when:

- it contains at least 10 chars
- it contains at least one lowercase letter
- it contains at least two digits
- it contains at least one of the following specials `!@#$%&*()`
- it does not contain the following specials `-_=+[]{}<>:?`

Note: Passwords in step1 should not be generated with the above constraints

Grading scale

Grading will be assessed according to:

- The respect of the rules
- Code Clarity: is it easy for one to understand the code
- Reproducibility: is it easy for one to run and update the code
- Packaging
- Documentation
- Extra Points: Unit testing