Assginment 1: using pure js (no built in function)

Using javascript answers the folllowing questions :

** Create a function that takes the age in years and returns the age in days.

** Create a function that takes an array of numbers and returns the smallest number in the set.

** Create a function that takes any non-negative number as an array and return it with its digits in descending order. Descending order is when you sort from highest to lowest.

** Create a function that takes an array of numbers and returns the average of this numbers.

** what is the output of
Console.log([ ] == [ ])
Console.log({ } == { })
And explain your answer

** what is the output of this code with explaination

```
function main() {
  console.log("A");
  setTimeout(function print() {
    console.log("B");
  }, 0);
  console.log("C");
}
main();
```

** what is the output of this code with explaination

```
var num = 8;
var num = 10;
console.log(num);
```

** what is the output of this code with explaination

```
function sayHi() {
  console.log(name);
  console.log(age);
  var name = 'Ayush';
  let age = 21;
}
sayHi();
```

Assignment 2:

using http*
* Create two arrays (user array, post array)

User end points
1- GetAllUsers
2- AddUser
3- Get all users sorted alphabetically by name
4- delete user
5- update user
6- search user by id

Post end points
1- GetAllPosts
2- AddPost
3- Get all Posts reversed (but don't change the order of the main array)
4- delete post
5- update post
6- search post by id

****

using Express*
* Create two arrays (user array, post array)

User end points
1- GetAllUsers
2- AddUser
3- Get all users sorted alphabetically by name
4- delete user
5- update user
6- search user by id

Post end points
1- GetAllPosts
2- AddPost
3- Get all Posts reversed (but don't change the order of the main array)
4- delete post
5- update post
6- search post by id

using mongoose and express and Joi
create two collections
*user collection * (userName, email , password ,age,gender , phone)
post collection (title,content , userID => must be related to the user collection)

*_user APIs_*
1-sign up (email must be unique)
2-sign in
3-update user use(findByIdAndUpdate)
4-delete user use(findByIdAndDelete OR deleteOne)
5-search for user where his name start with "*X*" and age less than *Y*=>   *(X,Y => variables)*
6-search for user where his age is between X and Y
7-get all user
8- get user profile with user posts(using populate)

*_Post APIs_*
1- add post *(make sure that user already exist)*
2- delete post *(post creator only)*
3- update post *(post owner only)*
4- get all posts
5- get all posts with their owners informaion _(using populate)_
6- sort posts descending *(By date)*