The following exercises are related to the use of Typescript.

1. Create a git repository for your answers to this problem sheet. Push the repository to GitHub. Make a commit and push it to GitHub after each exercise.

Solution:

```
mkdir answers-TypeScript
------------------------------------------------------------
cd answers-TypeScript
------------------------------------------------------------
echo > README.md
------------------------------------------------------------
git init
Initialized empty Git repository in /Users/martin/answers-TypeScript/.git/
------------------------------------------------------------
git add .
------------------------------------------------------------
git commit -m "Empty first commit."
[master (root-commit) 6583cf6] Empty first commit.
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
------------------------------------------------------------


git remote add origin https://github.com/mkenirons/WHATEVER.git
You will need to add the path to your GitHub Repository here
------------------------------------------------------------
git push -u origin master
Add you file to GitHub (remote repository)
------------------------------------------------------------
```

2. In this exercise you are required to create an array, and then create functions to add, remove and display the items in this array. To complete this exercise you will need to do the following:

   (a) Create an array of strings.

   (b) Create an addTask function:

      i. It receives a string as a parameter called task.

      ii. It adds the task to the array.

      iii. It prints a message in the console indicating the insertion.

      iv. It returns the number of elements in the array after the insertion.

   (c) Create a listAllTasks function:

   i. It iterates over all the tasks in the array.
   ii. It prints each array item in the console.

(d) Create a deleteTask function:
   i. It receives a string as a parameter called task.
   ii. It removes that string from the array.
   iii. It prints in console a message indicating the deletion.
   iv. It returns the number of elements in the array after the deletion.

Solution:

```
/ A. Create an array of strings
let tasks: Array<string> = [];
console.log("");
// B. Create a function to add a task in to the array, call it `addTask`,
//the function has to accept a string and doesn't have to return anything.
function addTask(task: string): number {
    let y: number;
    tasks.push(task);
    console.log("=========== NEW TASK ===========");
    console.log('Task "'+task+'" inserted in the list');
    y=tasks.length;
    return y;
}
// C. Create a function to list all tasks, it must show in the console the task.
function listAllTasks(){
    console.log("");
    console.log("START: Items on list");
    console.log("");
    tasks.forEach(function(task) {
        console.log(task);
    });
    console.log("");
    console.log("END: Items on list");
    console.log("");
}
// D. Create a function to delete a task, you must find the task inside
// the array and delete it.
function deleteTask(x: string): number{
    let key: string = x;
    let index: number = tasks.indexOf(key, 0);
    //https://www.tutorialspoint.com/typescript/typescript_array_indexof.htm
    if (index > -1) { // if >-1 this implies a match was found.
        tasks.splice(index, 1);
        //https://www.tutorialspoint.com/typescript/typescript_array_splice.htm
    }
    console.log("=========== TASK REMOVED ===========");
    console.log('Task "'+x+'" removed from the list');

    return tasks.length;
}
// Execution
console.log("Number of items:", addTask('This is our first task'));
console.log("Number of items:", addTask('This is our second task!!!'));
listAllTasks();
console.log("Number of items:", deleteTask('This is our second task!!!'));
listAllTasks();
```

3. In this exercise, the aim is to implement the same functionality as exercise 2 using an interface instead. A class will be required to implement the interface. To complete this exercise you will need to do the following: Create an interface and add the following to it:

   (a) An array of strings.

   (b) An addTask function which:

      i. Recieves a string as a parameter.

      ii. Prints a message in the console indicating the insertion.

      iii. Returns the number of elements in the array after the insertion.

   (c) A listAllTasks function which:

      i. Prints in the console all the tasks of the list.

   (d) A deleteTask function which:

      i. Recieves a string as a parameter.

      ii. Prints a message in the console indicating the deletion.

      iii. Returns the number of elements in the array after the deletion.

Solution:

```
// File todoInterface
// Write the interface for class Todo. It must have:
// - An array of strings as a property.
// - Method for adding task where will receive a string, doesn't return anything.
// - Method for listing all task in the console, doesn't return anything.
// - Method for delete a task where will receive a string, doesn't return anything.
// - You will need to export the inferface so it is available for import elsewhere.

export interface todoInterface{
    tasks: Array<string>;
    addTask(x: string): number;
    listAllTasks();
    deleteTask(x: string): number;
}

//Class in something.ts

import {todoInterface} from './todoInterface';

class Todo implements todoInterface{
    constructor(){}

    tasks: Array<string>=[];

    addTask(x: string): number{
        let y: number;
        this.tasks.push(x);
        console.log("=========== NEW TASK ===========");
        console.log('Task "'+x+'" inserted in the list');
        y=this.tasks.length;
        return y;
    }

    listAllTasks(){
        console.log("");
        console.log("START: Items on list");
        console.log("");
        this.tasks.forEach(function(task) {
            console.log(task);
        });
        console.log("");
        console.log("END: Items on list");
        console.log("");
    }

    deleteTask(x: string): number{
        let key: string = x;
        let index: number = this.tasks.indexOf(key, 0);
        if (index > -1) {
            this.tasks.splice(index, 1);
        }
        console.log("=========== TASK REMOVED ===========");
        console.log('Task "'+x+'" removed from the list');

        return this.tasks.length;
    }
}

let myTodos = new Todo();
console.log("Number of items:", myTodos.addTask('This is our first task'));
console.log("Number of items:", myTodos.addTask('This is our second task!!!'));
myTodos.listAllTasks();
console.log("Number of items:", myTodos.deleteTask('This is our second task!!!'));
myTodos.listAllTasks();
```