1) Given the following JavaSript programs, write the output for each of the following. (Each part is to be treated independently.)

a)

```
function printMe() {
    for (let j = 1; j < 4; j++)
        console.log("My name is Dodo");
}

printMe();
console.log("End of program");
```

Output:

b)

```
function printMe() {
    for (let i=1; i<3; i++){
        console.log("\nRound: " + i)
        for (let j = 1; j < 4; j++)
            console.log("My name is Dodo");
    }
}
printMe();
console.log("\nEnd of program");
```

Output:

c) Fill in the blanks to produce the required output below.

```
function printMe() {
    for (let i=1; ______________________){
        console.log("\nRound: " + i)

        for ( let j=0; ________________________)
            console.log("My name is Dodo");
    }
}
printMe();
console.log("\nEnd of program");
```

Output:

```
Round: 1
My name is Dodo
My name is Dodo

Round: 2
My name is Dodo
My name is Dodo

Round: 3
My name is Dodo
My name is Dodo
```

d) What is the output for the following JavaScript codes?

```
function printMe() {
    for (let i=1; i<3; i++){
        console.log("\nRound: " + i)
        for (let j = 1; j < 3; j++){
            let genNum = Math.floor(Math.random( ) * 9);
            console.log("Number generated is : " + genNum);
        }
    }
}
printMe();
console.log("\nEnd of program");
```

Output:

2) What is the output?

a)

```
let myArr = [1,2,3,4];
let i=1;
let s='';

while (i<=2){
    for (let j=0; j < myArr.length; j++){
        s = s + (i*myArr[j]) + " ";
    }
    i = i + 1;
    s = s + "\n";
}
console.log(s);
```

Output:

b) Which line would you change in the above program to print one more row of numbers? What is the output of the 3rd row?

c) Open your Visual Studio, rewrite the codes in Question 2(a) using function.

Write a function **printNumbers(noRows, arr)** which will:

- accept two parameters ie array of numbers and required number of rows to be printed
- use nested loops to create and store the rows of numbers in a local variable **s** [hint : use the codes in Question 2(a)]
- returns content of **s** to the main program for output display

In the main program :

- prompts the user for the number of rows to be printed
- no data validation is required, you may assume that user always enters the correct data
- invoke function **printNumbers** by passing in the required number of rows to be printed and the array **myArr**
- display on the console the content that is returned from the function

```
let input = require('readline-sync');
let myArr = [1,2,3,4];
let requiredRows = parseInt(input.question('Please enter
rows needed:'));

//Complete the function below
function printNumbers(noRows,arr){



}
// Invoke function here
```

d) Amend program in part (c) to include data validation and prompt user if he/she wish to continue. You may assume that if the user were to enter number, it will always be an integer. The program should allow only a maximum of 10 rows to be displayed each time.

Sample output, text in **bold** are user's data input.

```
Please enter rows needed: 0
Please enter a valid number for rows needed.
Please enter rows needed: -5
Please enter a valid number for rows needed.
Please enter rows needed: x
Please enter a valid number for rows needed.
Please enter rows needed: 11
Please enter a valid number for rows needed.
Please enter rows needed: ?
Please enter a valid number for rows needed.
Please enter rows needed: 4
1 2 3 4
2 4 6 8
3 6 9 12
4 8 12 16

Enter 1 to continue, 0 to quit : 1
Please enter rows needed: 10
1 2 3 4
2 4 6 8
3 6 9 12
6 12 18 24
7 14 21 28
8 16 24 32
9 18 27 36
10 20 30 40

Enter 1 to continue, 0 to quit : 0
Goodbye, program terminated!
```

3) Modify Practical 6a Q8, ***Prize Money*** program.

a) Write a function ***getInput*** that prompts for the rank of the contestant and returns the value entered. The function does not take in any parameter and returns an integer value.

b) Write a function ***printPrize*** that takes in ***rank*** as its parameter. It checks and displays the prize money based on the rank. The function does not return any value.

c) Invoke the ***getInput*** and ***printPrize*** functions to test the program.

4a) Run the program below and what do you observe about the output format? What do you notice about process.stdout.write?

```
console.log("Output 4 numbers generated using random
function.");
for (let j = 1; j <= 4; j++) {  //4D ie 4 columns
        digit = Math.floor(Math.random() * 10);
        process.stdout.write(digit + " ")
```

4b) Write a program, **using nested loop**, that generates 5 sets of 4 single-digit numbers (0 to 9) randomly as follows:

```
9 3 3 9
3 9 8 8
2 3 7 8
0 1 6 6
7 5 2 5
```

4c) [Optional] You may want to try using function to implement the above, prompt user for number of rows and columns to display and do some data validation. Note : Solution are not available, do a complete test plan to ensure your program runs properly.

5) [Optional] Modify Practical 6a Q3 using function.

a) Write an integer function ***getInput*** as follows:

```
function getInput(s)
```

The function takes in a String parameter that will be used as part of the input prompt, e.g. "Please enter **1st** integer" and "Please enter **2nd** integer". It returns an integer value corresponding to the user input.

b) Write a function ***findMax*** that takes in 2 integer values ***num1*** and ***num2***. It compares which number is bigger and returns one of the following string values:

"1st number is bigger"
"2nd number is bigger"
"The 2 numbers are equal"

c) In the main part of the program, invoke ***getInput("1st")*** and ***getInput("2nd")*** to get the values of ***num1*** and ***num2***. Invoke ***findMax*** and display the string returned.

6) Complete the missing parts of the following program by writing the appropriate functions.

```
let input = require("readline-sync");

let pat = choosePattern();
let num = readInput();

switch(pat) {
    /*part f*/
}

function readInput() {
    /*part a*/
}

function choosePattern(){
    /*part b*/
}

function printPattern1(n){
    /*part c*/
}

function printPattern2(n){
    /*part d*/
}

function printPattern3(n){
    /*part e*/
}
```

a) Code the ***readInput()*** function to prompt the user for the number of rows. The function then returns the number entered.

```
Enter the number of rows: 4
```

b) Complete the ***choosePattern()*** function to prompt the user for the pattern he wish to be printed. The function returns the number entered to the calling code.

```
1. Print Pattern 1
2. Print Pattern 2
3. Print Pattern 3
4. Exit
> 3
```

c) Complete the ***printPattern1(n)*** function to print ***Pattern 1*** as shown (assume ***n*** is 4).

Pattern 1

```
1    1    1    1
2    2    2    2
3    3    3    3
4    4    4    4
```

d) Complete the ***printPattern2(n)*** function to print ***Pattern 2*** as shown (assume ***n*** is 4).

Pattern 2

```
1    2    3    4
1    2    3    4
1    2    3    4
1    2    3    4
```

e) Complete the ***printPattern3(n)*** function to print ***Pattern 3*** as shown (assume ***n*** is 4).

Pattern 3

```
1    2    3    4
2    4    6    8
3    6    9    12
4    8    12   16
```

f) Complete the switch statement based on the pattern selected to invoke ***printPatternX*** function coded to produce the output.

Sample output 1:

```
Enter the number of rows: 4
1. Print Pattern 1
2. Print Pattern 2
3. Print Pattern 3
4. Exit
> 1

1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
```

Sample output 2:

```
Enter the number of rows: 4
1. Print Pattern 1
2. Print Pattern 2
3. Print Pattern 3
4. Exit
> 3

1 2 3 4
2 4 6 8
3 6 9 12
4 8 12 16
```

[**Optional exercise :** Try adding a loop to enable user to continue printing until user select e.g 0 to exit program.]

7. **Objective**: A PE teacher wants to build a JavaScript program to manage a list of students, their genders, and heights. She requires the program to allow her to display list of students' name, add students, create teams for the boys and girls and calculate average height of all her students.

Table 1 is the incomplete JavaScript program. Read the instructions carefully and code carefully.

a) Complete the function **displayName** which:
- accepts one parameter i.e. array **studentName,**
- does not return any value, and
- displays all students' names in the function using a ***for*** loop. Do **NOT** hardcode.

Sample output, text in **bold** are user input

```
1. Display namelist for all students
2. Add a student
3. Team Formation Submenu
4. Find Average Height
0. Exit

Enter a choice 0-4: 1

Namelist of all students:
1. Tommy
2. Samuel
3. Joy
4. Ali
5. Lee Lee
6. Anne
7. David

Menu:
```

```
1. Display namelist for all students
2. Add a student
3. Team Formation Submenu
4. Find Average Height
0. Exit

Enter a choice 0-4:
```

b) Complete the function **addStudent** which:

- does not accept any parameter,
- prompts user input for name, gender and height of new record
- does not return any value, and
- add into the respective array.

Data validation is not required.

```
1. Display namelist for all students
2. Add a student
3. Team Formation Submenu
4. Find Average Height
0. Exit

Enter a choice 0-4: 2

Enter new student's name: Pauline
Enter gender of new student (M or F): F
Enter height: 158

Menu:
1. Display namelist for all students
2. Add a student
3. Team Formation Submenu
4. Find Average Height
0. Exit
```

c) Complete the function **teamFormation** which:

- does not accept any parameter,
- prompts user input for choices in sub menu
- does not return any value, and
- Use case_switch to invoke the respective function as below

Data validation is not required for submenu options.

User Input	Menu options	Function to invoke/Remarks
1	Create boys and girls teams	creatureTeams
2	Display namelist of a team	displayName
0	Exit to main menu	Main menu displayed

Complete the function **createTeams** which:

- does not accept any parameter,
- does not return any value, and

- create a girls and a boys team, store their names and height in 4 different arrays: **girlsName, girlsHeight, boysName and boysHeight.**

Complete the function **displayTeamPlayers** which:

- does not accept any parameter,
- does not return any value,
- prompt user to input gender
- display name of players based on gender

Sample output, text in **bold** are user input.

```
Menu:
1. Display namelist for all students
2. Add a student
3. Team Formation Submenu
4. Find Average Height
0. Exit

Enter a choice 0-4: 3

        Sub Menu:
        1. Create boys and girls teams
        2. Display namelist of a team
        0. Exit to main menu

Enter a choice 0-3: 1
Girls team:
1. Joy
2. Lee Lee
3. Anne

Boys team:
1. Tommy
2. Samuel
3. Ali
4. David

        Sub Menu:
        1. Create boys and girls teams
        2. Display namelist of a team
        0. Exit to main menu

Enter a choice 0-3: 2

Enter 'G' or girls team and 'B' for boys team for the namelist : G
1. Joy
2. Lee Lee
3. Anne

        Sub Menu:
        1. Create boys and girls teams
        2. Display namelist of a team
        0. Exit to main menu

Enter a choice 0-3: 0
Exiting to main menu...

Menu:
1. Display namelist for all students
2. Add a student
3. Team Formation Submenu
4. Find Average Height
0. Exit

Enter a choice 0-4:
```

d) Complete the function **findAverage** which:

- accept one parameter, ie **studentHeight** array
- return the average height of all the students to be displayed in the main program

```
Menu:
1. Display namelist for all students
2. Add a student
3. Team Formation Submenu
4. Find Average Height
0. Exit

Enter a choice 0-4: 4
Average height of all students: 1.61m

Menu:
1. Display namelist for all students
2. Add a student
3. Team Formation Submenu
4. Find Average Height
0. Exit

Enter a choice 0-4: 0
Thank you & goodbye!
// program terminates
```

Table 1

```
const readline = require('readline-sync');

const studentName = ["Tommy", "Samuel", "Joy", "Ali", "Lee Lee", "Anne",
"David"];
const studentGender = ["M","M","F","M","F","F","M"];
const studentHeight = [174,170,160,165,165,151,140];

let girlsName = [];
let girlsHeight = [];
let boysName = [];
let boysHeight = [];

function displayMenu() {
    console.log("\nMenu:");
    console.log("1. Display namelist for all students");
    console.log("2. Add a student");
    console.log("3. Team Formation Submenu");
    console.log("4. Find Average Height");
    console.log("0. Exit");
}



function displaySubmenu() {
    console.log("\n\tSub Menu:");
    console.log("\t1. Create boys and girls teams");
    console.log("\t2. Display namelist of a team");
    console.log("\t0. Exit to main menu");
}
//

function displayName(arrlist){


}
function addStudent() {


}
function findAverage(heightArr) {


}

function createTeams() {
}

//3. Display Team Formation submenu
function teamFormation() {
let subChoice = '';

    while (subChoice !== 0) {
        displaySubmenu();
          :
          :
}


//----------------------------------
//main program

let result = true;
while (result) {
      displayMenu();
      // continue coding



} // end while
```

- END -