Lab-3 Question

Time: 90 min **Maximum Marks: 60**

General Instruction:

- Binary marking will be there that is 20 or 0 marks.
- If your code is successfully compiled and the desired output is produced, then only we will check for the code otherwise, 0 marks.
- No request for late submission will be entertained in any circumstances.
- Submit all your codes using the proper naming convention (read the instructions for each question) and put all files in a folder.
- Naming Convention: “Name_Roll Number.zip”, and upload it.

***Note: Use bash shell for each shell scripting question.**

Question 1: Write a shell script program to print the first 5 lines of a File **(Marks: 10)**

Submission Guidelines:

1. **File Naming:** Save your .sh program as `first_05_Line.sh`
2. `If file contains the fewer than 5 lines, then print all lines.`
3. `Your script will be executed as follows:`

```
./first_05_Line.sh <file_name>
```

Question 2: Write a shell script program to count the occurrences of a given word in a file. **(Marks: 10)**

Submission Guidelines:

4. **File Naming:** Save your .sh program as count_word.sh
5. If a word is not present in the file then output should be 0.
6. Your script will be executed as follows:

```
./count_word.sh <file_name> <word>
```

Question 3: Write a C program to manage a dynamic array of integers. (Marks: 10)

The program should perform the following tasks:

1. Create a Dynamic Array
2. Populate the Array
3. Display the Array
4. Resize the Array
5. Reverse the Array
6. Free the Memory

Example:

Enter the size of the array: 5
Enter 5 elements: 1 2 3 4 5
The array elements are: 1 2 3 4 5
Do you want to resize the array? (y/n): y
Enter the new size of the array: 7
Enter 2 more elements: 6 7
The array elements after resizing are: 1 2 3 4 5 6 7
Reversing the array...
The array elements after reversing are: 7 6 5 4 3 2 1
Freeing the memory and exiting the program…

Submission guidelines:

File Naming: Save your single C source file as dynamic_array.c

Question 4: You are given 2 C programs (a.c and b.c), each of them prints a list of 1000 no's on standard output. Write a shell script that executes both

the programs and find the first index where the output generated a.c and b.c differs. (Marks: 30)