Question 1:

Write a Bash script that:

1. Declares an **array** of five numbers.
2. Declares a **normal variable** called threshold with a numeric value.
3. Iterates over the array using a for loop.
4. Inside the loop, checks if the current number is greater than the threshold.
    - If the number is greater, print "X is greater than threshold" (replace X with the actual number).
    - Otherwise, print "X is not greater than threshold".

Example Output:

If the array is (10 25 7 30 15) and threshold=15, the script should output:

```
10 is not greater than threshold
25 is greater than threshold
7 is not greater than threshold
30 is greater than threshold
15 is not greater than threshold
```

Bonus:

Modify the script to allow the user to input the threshold value instead of hardcoding it.

Question 2:

Write a Bash script that:

1. Declares a **global variable** called log_file and sets it to "system.log".
2. Declares a **function** called process_logs that:
    - Uses a **local variable** to store the count of matching lines.
    - Takes a **regular expression** as an argument.
    - Uses grep with the provided regex to count matching lines in system.log using $(...).
    - Prints how many lines match the given pattern.
3. Uses a **for loop** to test multiple patterns on the log file.
4. Uses an **if statement** inside the function to check if any matches were found:
    - If matches exist, print "Found X matching lines for pattern 'Y'".
    - Otherwise, print "No matches found for pattern 'Y'".

Example system.log Content:

```
[ERROR] Disk space is low
[INFO] System update completed
[WARNING] High CPU usage detected
[ERROR] Failed to connect to database
[INFO] User logged in
```

Example Execution & Output:

If the script runs with patterns ERROR, INFO, and DEBUG:

```
Found 2 matching lines for pattern 'ERROR'
Found 2 matching lines for pattern 'INFO'
No matches found for pattern 'DEBUG'
```

Bonus:

Modify the script to allow the user to enter custom regex patterns instead of using hardcoded ones.