Homework #3: Shell Script Programming

Script 1. Backup Automation Script

Context:

You work as a system engineer for a company that manages critical project data stored in various directories. The backup process is currently manual, prone to errors, and takes a lot of time. Your manager asks you to automate the backup process using a shell script.

Problem:

Write a shell script that:

1. **Accepts three parameters:**
    - Source directory to back up (e.g., `/home/projects`)
    - Backup destination directory (e.g., `/mnt/backup`)
    - Maximum number of backups to keep (e.g., 5)
2. **Creates a compressed archive (`.tar.gz`) of the source directory,** named with the current date/time, and stores it in the backup destination.
3. **Maintains only the latest N backups** (as specified by the max backups parameter), deleting older backups beyond this number.
4. **Logs each backup action** (start time, end time, backup file name, success/failure) to a log file `/home/youruser/backup.log`.
5. **Checks for errors** such as:
    - Source directory doesn't exist
    - Destination directory not writable
    - Failure during compression or deletion

Requirements:

- Use variables, functions, and conditionals.
- Use commands like `tar`, `ls`, `rm`, `date`, `echo`.
- Provide clear user messages.
- Write comments explaining your code.

Deliverables:

- Shell script file named `auto_backup.sh`.
- README file with instructions on usage, examples, and assumptions.

Bonus (optional):

- Add an option to send an email notification upon backup completion.
- Include a feature to exclude certain file types or directories from the backup.

Example Usage:

./auto_backup.sh /home/projects /mnt/backup 5

Script 2. Log File Analysis and Alerting

Context:

You are a system administrator responsible for monitoring application logs on a Linux server. The application writes logs to a specific file, and errors need to be detected quickly to respond proactively.

Problem:

Write a shell script that:

1. **Accepts as input the path to a log file** (e.g., `/var/log/app.log`).
2. **Scans the log file for any lines containing the word "ERROR" or "WARNING".**
3. **Counts how many ERRORs and WARNINGs appeared in the last N lines of the log file** (default 1000 lines; allow this number as an optional parameter).
4. **Generates a summary report:**
    - Number of ERRORs found
    - Number of WARNINGs found
    - Timestamp of the most recent ERROR or WARNING
5. **If the count of ERRORs exceeds a threshold (default 10), output a critical alert message.**
6. **Save the report to a file named `log_alert_report.txt` in the current directory.**
7. **Gracefully handle errors such as:**
    - Log file not found or unreadable
    - No ERROR/WARNING entries found

Requirements:

- Use command-line utilities like `grep`, `tail`, `wc`, `awk`, and shell scripting constructs.
- Use parameters and defaults.
- Provide clear messages and comments.
- Format the report neatly.

Deliverables:

- Shell script named `log_monitor.sh`
- README with usage instructions and examples

Bonus (optional):

- Add an email notification feature if the critical alert triggers.
- Include the ability to specify different keywords to monitor.

Example Usage:

```
./log_monitor.sh /var/log/app.log 2000
```

Sample Log File (app.log):

```
2025-06-08 08:00:01 INFO Starting application version 1.2.3
2025-06-08 08:00:05 WARNING Disk space low on /dev/sda1
2025-06-08 08:00:10 INFO User admin logged in
2025-06-08 08:01:00 ERROR Failed to connect to database
2025-06-08 08:02:30 INFO Scheduled job started
2025-06-08 08:03:45 WARNING High memory usage detected
2025-06-08 08:04:00 INFO Scheduled job finished successfully
2025-06-08 08:05:12 ERROR Timeout while contacting external API
2025-06-08 08:10:00 INFO Backup process initiated
2025-06-08 08:15:15 INFO Backup process completed
2025-06-08 08:20:30 ERROR User authentication failed for user
guest
2025-06-08 08:25:00 INFO User admin logged out
2025-06-08 08:30:05 WARNING Network latency above threshold
2025-06-08 08:35:45 INFO System health check passed
2025-06-08 08:40:10 ERROR Disk read error on /dev/sdb2
2025-06-08 08:45:30 INFO User backup started
2025-06-08 08:50:00 INFO User backup completed
2025-06-08 08:55:20 WARNING CPU temperature high
2025-06-08 09:00:00 INFO Application shutdown initiated
```