# Experiment 12: Building a Rule-Based Expert System

Name: Sankalp Ronge

Roll No.: 590027268

Date: 2025-11-30

## Aim

To develop a rule-based expert system using shell scripting, and automate system tasks using cron scheduling and system administration scripts.

## Requirements

- Linux system with Bash shell

- Text editor (nano/vim)

- Access to cron

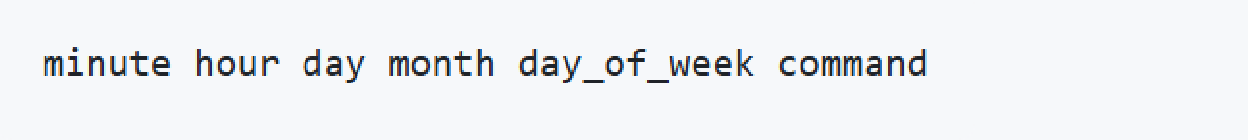
- Basic shell scripting knowledge

## Theory

### 1. Process Automation & Job Scheduling

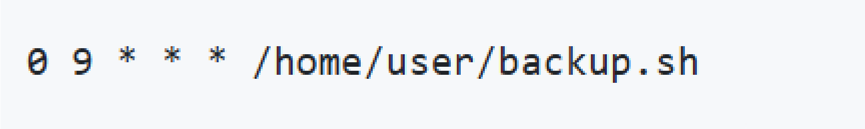
Linux provides tools to automate tasks:

#### cron (Recurring tasks)



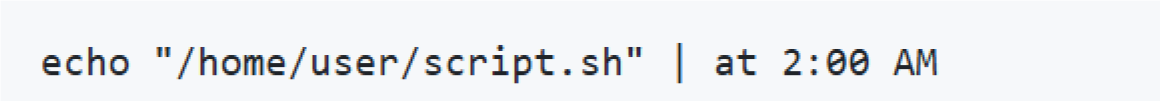
~~~
minute hour day month day_of_week command
~~~

Example:



~~~
0 9 * * * /home/user/backup.sh
~~~

```

#### at (One-time tasks)

```



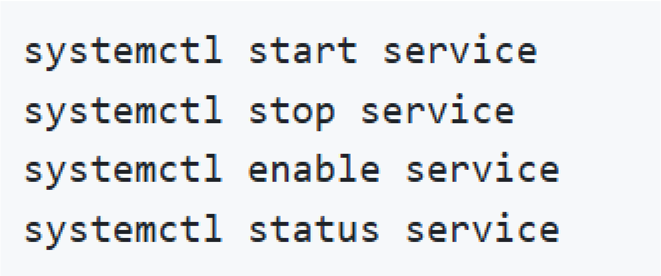
~~~
echo "/home/user/script.sh" | at 2:00 AM
~~~

```

---

### 2. System Administration Scripts

Used for automated:

- Backups

- User management

- Log monitoring
```

- Health checks

### 3. Services and Daemons

Managed using systemctl:



```
systemctl start service
systemctl stop service
systemctl enable service
systemctl status service
```

```

```

## Procedure & Observations

Students created a rule-based medical expert system using:

- If-else logic

- Pattern matching

- Multi-symptom handling

- Cron scheduling

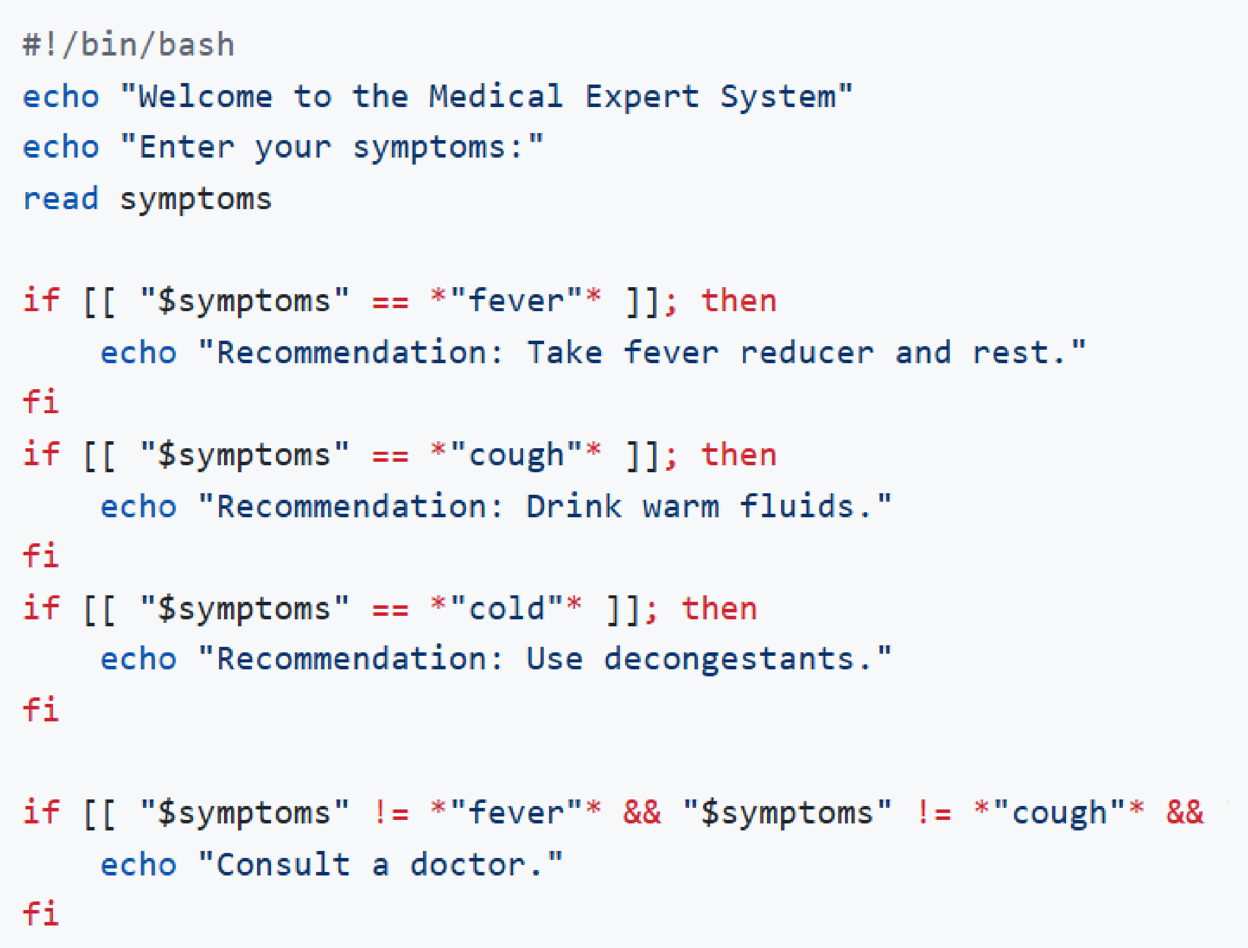
# LAB EXERCISE

## Exercise: Basic Medical Expert System

### Task Statement

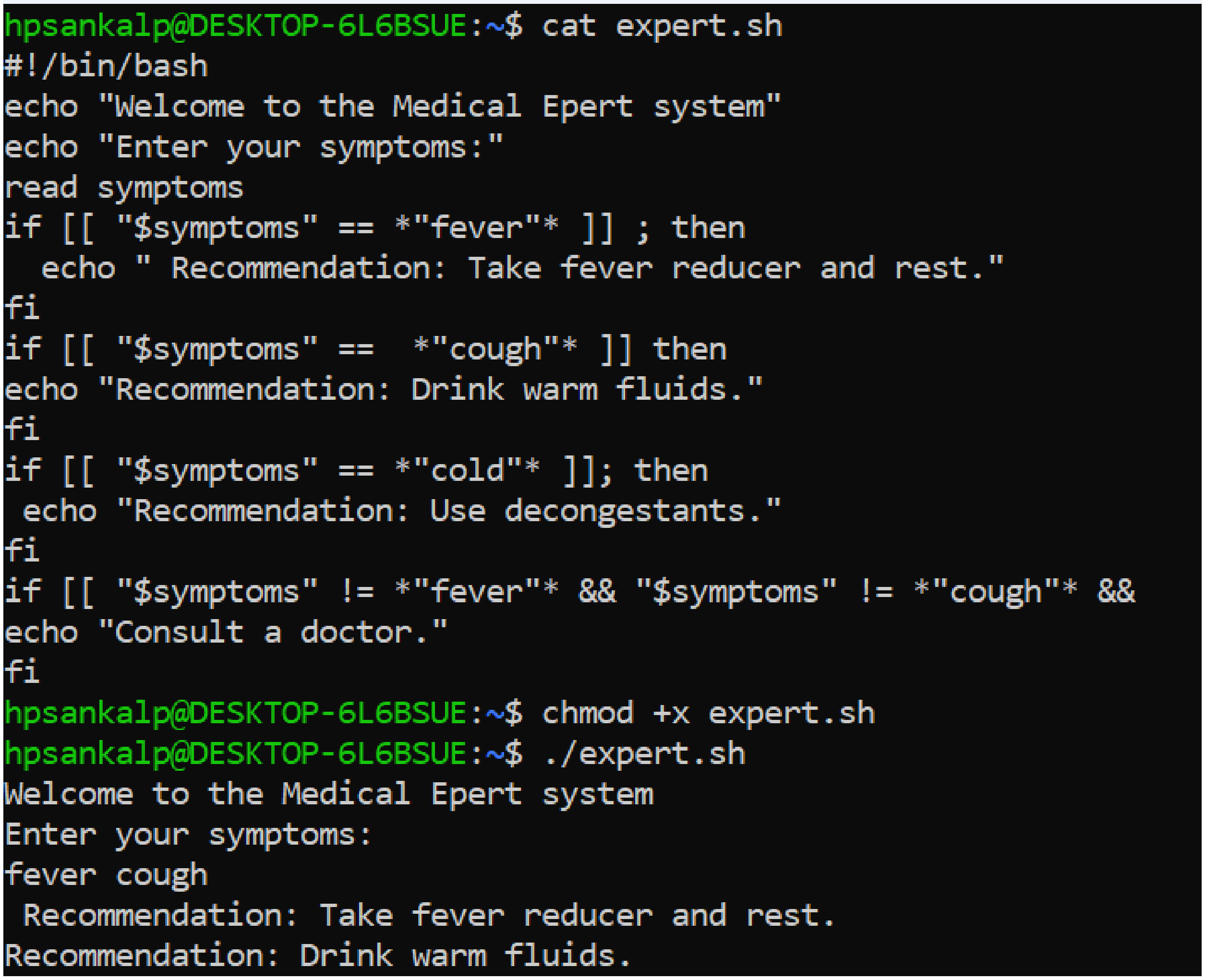
Create a simple expert system that gives recommendations based on user symptoms.

### Commands



```
#!/bin/bash
echo "Welcome to the Medical Expert System"
echo "Enter your symptoms:"
read symptoms

if [[ "$symptoms" == *"fever"* ]]; then
    echo "Recommendation: Take fever reducer and rest."
fi
if [[ "$symptoms" == *"cough"* ]]; then
    echo "Recommendation: Drink warm fluids."
fi
if [[ "$symptoms" == *"cold"* ]]; then
    echo "Recommendation: Use decongestants."
fi

if [[ "$symptoms" != *"fever"* && "$symptoms" != *"cough"* &&
    echo "Consult a doctor."
fi
```

### Output



```
hpsankalp@DESKTOP-6L6BSUE:~$ cat expert.sh
#!/bin/bash
echo "Welcome to the Medical Epert system"
echo "Enter your symptoms:"
read symptoms
if [[ "$symptoms" == *"fever"* ]] ; then
  echo " Recommendation: Take fever reducer and rest."
fi
if [[ "$symptoms" ==  *"cough"* ]] then
echo "Recommendation: Drink warm fluids."
fi
if [[ "$symptoms" == *"cold"* ]]; then
 echo "Recommendation: Use decongestants."
fi
if [[ "$symptoms" != *"fever"* && "$symptoms" != *"cough"* &&
echo "Consult a doctor."
fi
hpsankalp@DESKTOP-6L6BSUE:~$ chmod +x expert.sh
hpsankalp@DESKTOP-6L6BSUE:~$ ./expert.sh
Welcome to the Medical Epert system
Enter your symptoms:
fever cough
 Recommendation: Take fever reducer and rest.
Recommendation: Drink warm fluids.
```

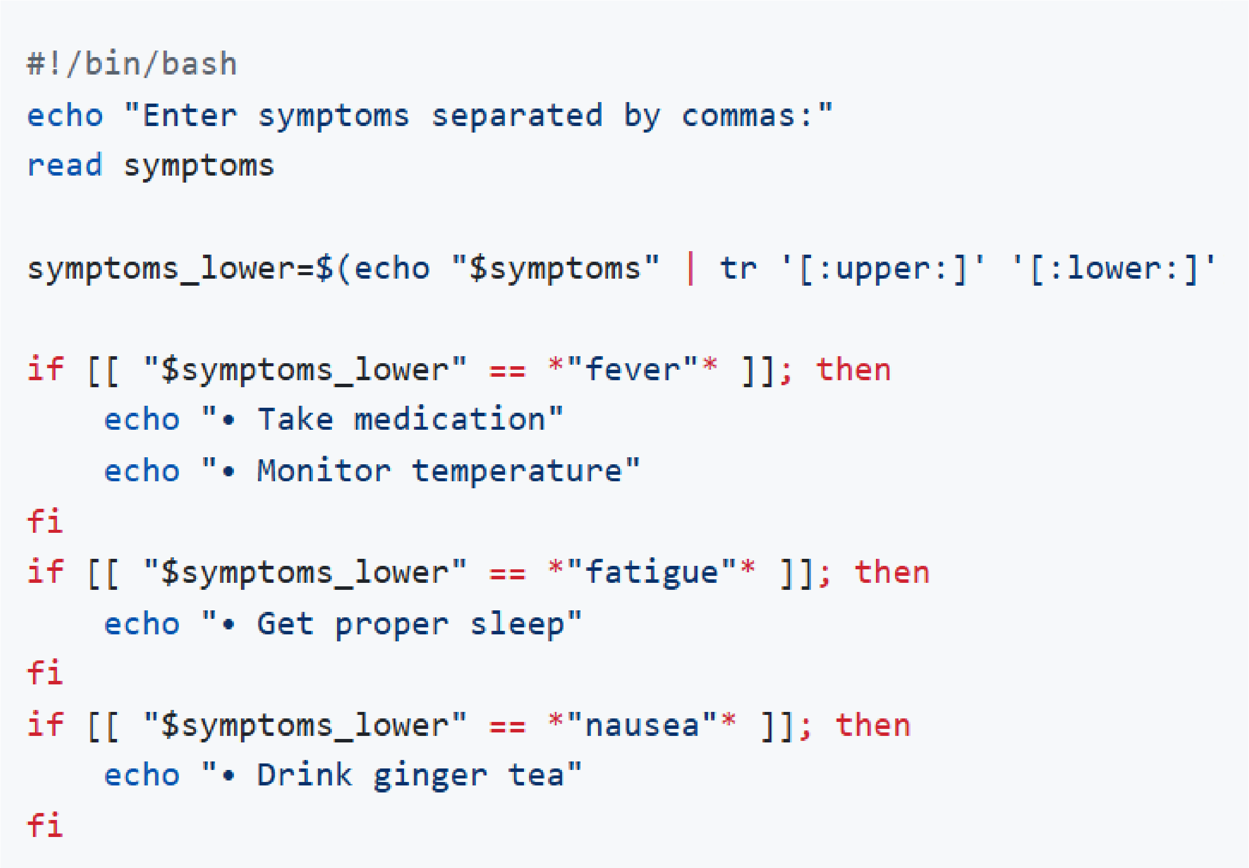
# ASSIGNMENTS

## Assignment 1: Extended Expert System

### Task Statement

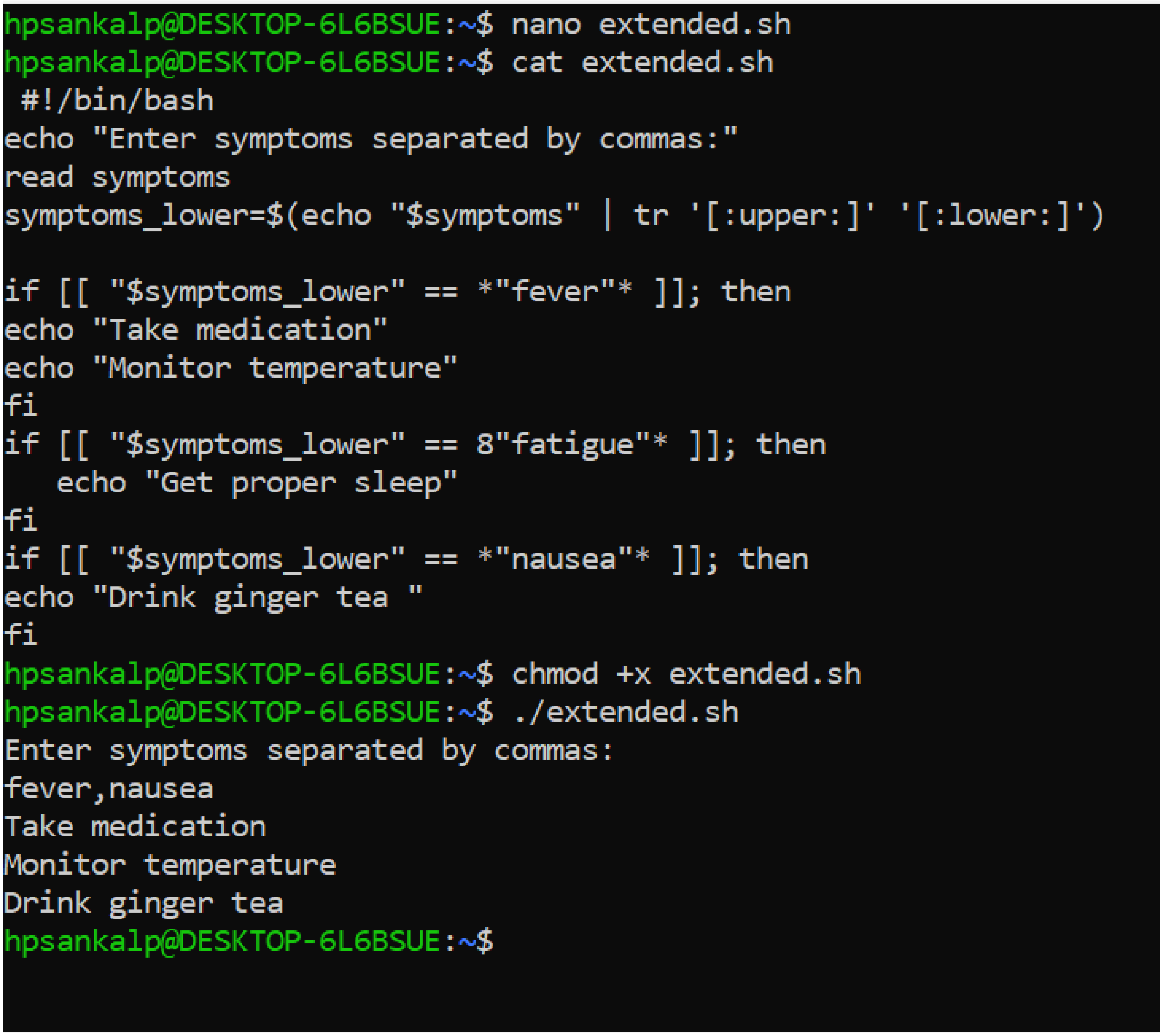
Improve the expert system with multiple symptoms and detailed recommendations.

### Commands



```bash
#!/bin/bash
echo "Enter symptoms separated by commas:"
read symptoms

symptoms_lower=$(echo "$symptoms" | tr '[:upper:]' '[:lower:]'

if [[ "$symptoms_lower" == *"fever"* ]]; then
    echo "• Take medication"
    echo "• Monitor temperature"
fi
if [[ "$symptoms_lower" == *"fatigue"* ]]; then
    echo "• Get proper sleep"
fi
if [[ "$symptoms_lower" == *"nausea"* ]]; then
    echo "• Drink ginger tea"
fi
```

### Output



```
hpsankalp@DESKTOP-6L6BSUE:~$ nano extended.sh
hpsankalp@DESKTOP-6L6BSUE:~$ cat extended.sh
 #!/bin/bash
echo "Enter symptoms separated by commas:"
read symptoms
symptoms_lower=$(echo "$symptoms" | tr '[:upper:]' '[:lower:]')

if [[ "$symptoms_lower" == *"fever"* ]]; then
echo "Take medication"
echo "Monitor temperature"
fi
if [[ "$symptoms_lower" == 8"fatigue"* ]]; then
   echo "Get proper sleep"
fi
if [[ "$symptoms_lower" == *"nausea"* ]]; then
echo "Drink ginger tea "
fi
hpsankalp@DESKTOP-6L6BSUE:~$ chmod +x extended.sh
hpsankalp@DESKTOP-6L6BSUE:~$ ./extended.sh
Enter symptoms separated by commas:
fever,nausea
Take medication
Monitor temperature
Drink ginger tea
hpsankalp@DESKTOP-6L6BSUE:~$
```

## Assignment 2: Scheduling Using Cron

### Task Statement

Schedule the expert system to run daily at 8 AM.

### Commands

```
crontab -e
```

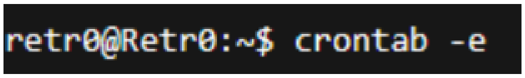
Add:

```
0 8 * * * /path/to/expert_system.sh
```

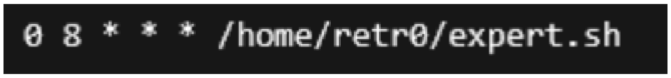
Make executable:

```
chmod +x /path/to/expert_system.sh
```

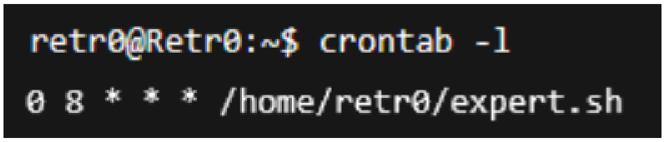
### Output



```
retr0@Retr0:~$ crontab -e
```



```
0 8 * * * /home/retr0/expert.sh
```



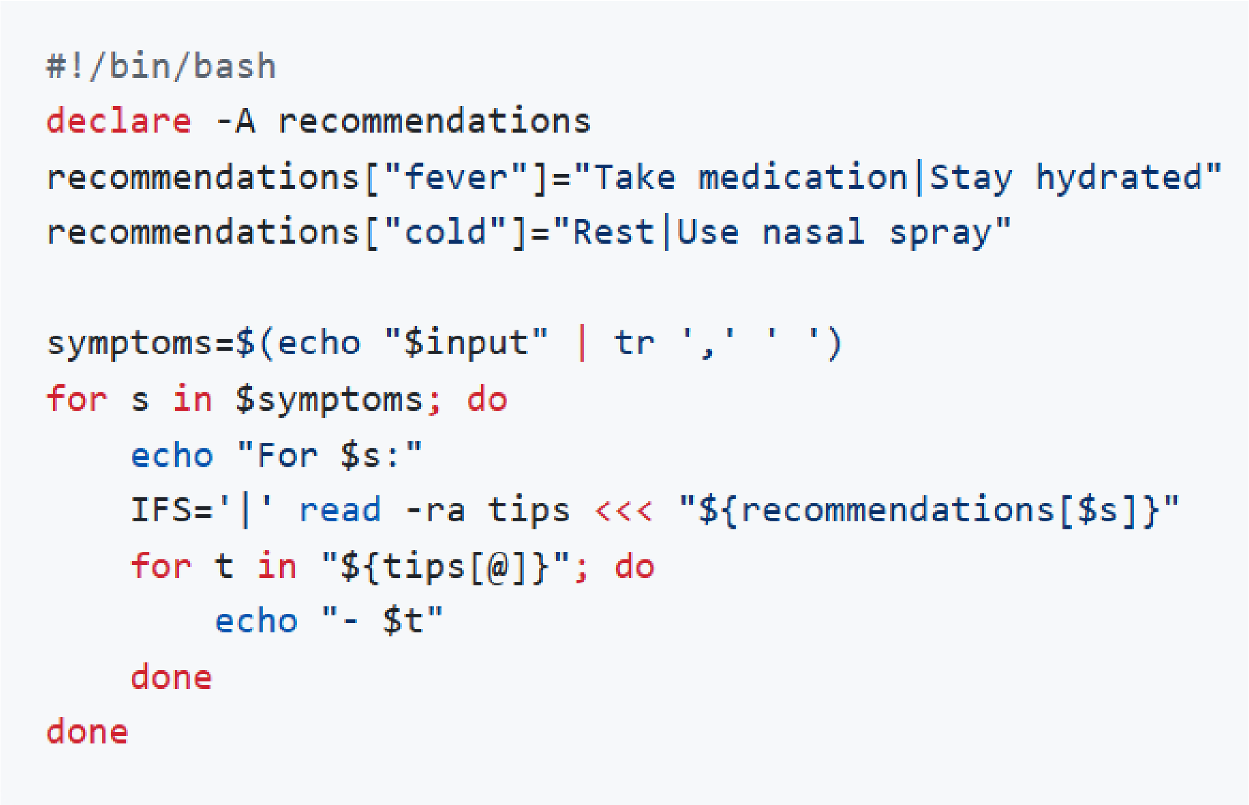
```
retr0@Retr0:~$ crontab -l
0 8 * * * /home/retr0/expert.sh
```

## Assignment 3: Multi-Recommendation System

### Task Statement

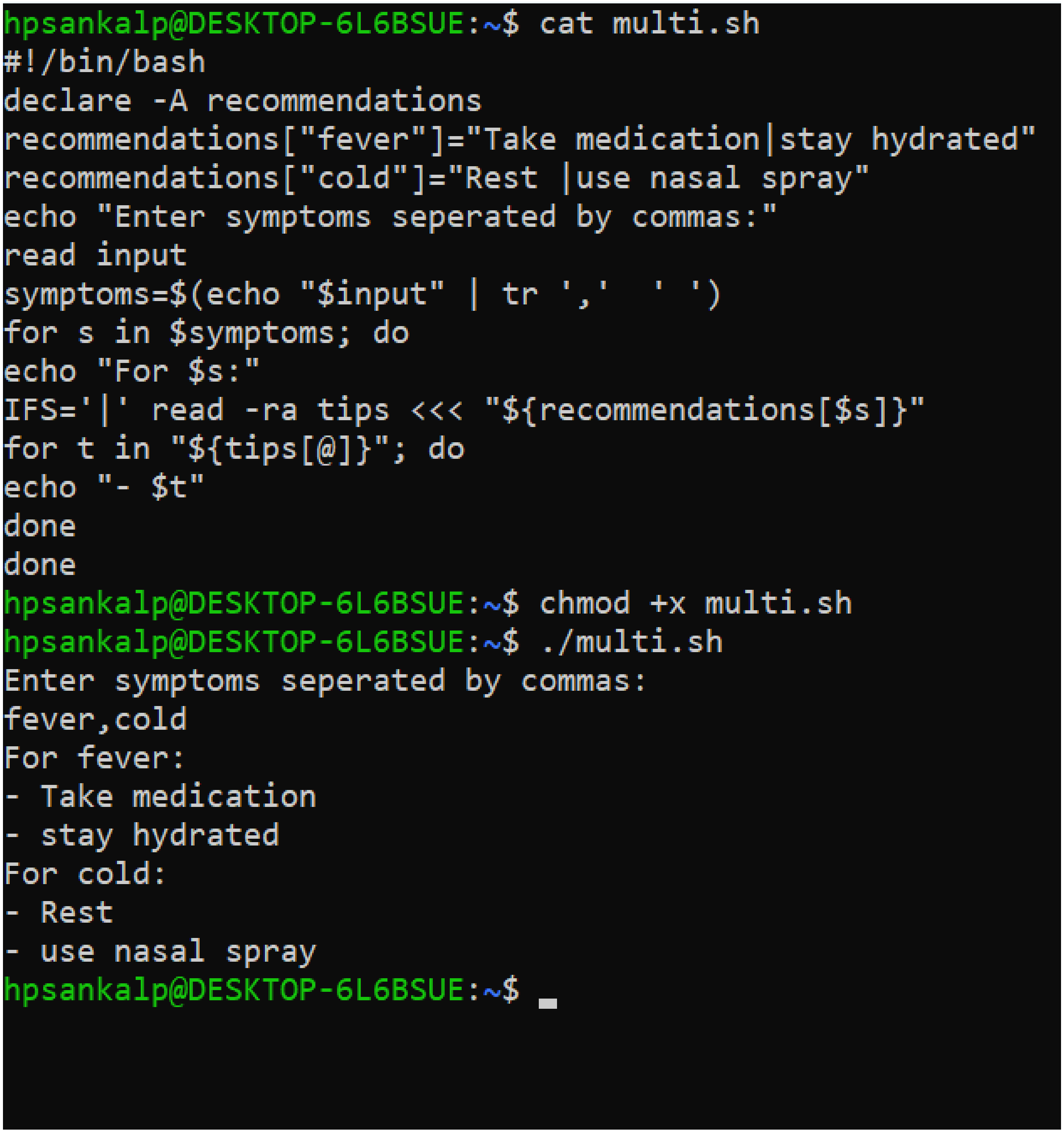
Use associative arrays to provide structured recommendations.

### Commands



```bash
#!/bin/bash
declare -A recommendations
recommendations["fever"]="Take medication|Stay hydrated"
recommendations["cold"]="Rest|Use nasal spray"

symptoms=$(echo "$input" | tr ',' ' ')
for s in $symptoms; do
    echo "For $s:"
    IFS='|' read -ra tips <<< "${recommendations[$s]}"
    for t in "${tips[@]}"; do
        echo "- $t"
    done
done
```

### Output



```
hpsankalp@DESKTOP-6L6BSUE:~$ cat multi.sh
#!/bin/bash
declare -A recommendations
recommendations["fever"]="Take medication|stay hydrated"
recommendations["cold"]="Rest |use nasal spray"
echo "Enter symptoms seperated by commas:"
read input
symptoms=$(echo "$input" | tr ','  ' ')
for s in $symptoms; do
echo "For $s:"
IFS='|' read -ra tips <<< "${recommendations[$s]}"
for t in "${tips[@]}"; do
echo "- $t"
done
done
hpsankalp@DESKTOP-6L6BSUE:~$ chmod +x multi.sh
hpsankalp@DESKTOP-6L6BSUE:~$ ./multi.sh
Enter symptoms seperated by commas:
fever,cold
For fever:
- Take medication
- stay hydrated
For cold:
- Rest
- use nasal spray
hpsankalp@DESKTOP-6L6BSUE:~$
```

## Result

The rule-based expert system was successfully built.

Cron scheduling and automation concepts were demonstrated.

## Conclusion

This experiment taught rule-based logic, automation, and expert system development using shell scripting.