Name:- Y. Nithin Kumar
Reg no:- 2000031112
Sec:- 13

HOME ASSIGNMENT - 1

1) Make a memory utilization check script on Linux host, and if it's lower than 1GB throw an error "Minimum of 1GB RAM is required". Verify exit code with help of: echo $.

A:) The bash script that performs a memory utilization check on a Linux host and throws an error if available memory is lower than 1GB:

bash

```
MEM_TOTAL = $(grep MemTotal /proc/meminfo | awk
                '{print $2}')
MEM_TOTAL = $(grep MemAvailable /proc/meminfo | awk
                '{print $2}')
if [$ (MEM_AVAIL /1024)) - lt 1024]; then
  echo "Minimum of 1GB RAM is required"
  exit 1
else
  echo "Memory check passed".
  exit 0
fi
```

* We can verify the exit code of script by running and checking value of '$'

bash

./memory_check.sh

echo $?

2. Create a backup script:

It should have configurable parameters: backup directory, list of directories to backup, database name and user. Files should be archived using tar command and compressed. Then placed in backup directory with date tag. Database should be dumped and then archived using tar command and compressed. Then placed in backup directory with the date tag. Add this script to contab.

Modify script so it will make separate daily and weekly backups.

And rotation script so it will keep only last 7 digit daily backups and 4 last weekly backups.

A:)

```
BACKUP_DIR=/path/to/backup/directory
DIRECTORIES_TO_BACKUP=(/path/to/dir1 /path/to/dir2)
DATABASE_NAME=database_name
DATABASE_USER=database_user

DATE_TAG=$(date +%Y-%m-%d)
IS_WEEKLY=false

if [ "$(date +%u)" -eq 7 ]; then
  IS_WEEKLY=true
fi

#backup function
backup() {
  local dir_to_backup=$1
  local target_file=$2

  tar -czf $target_file $dir_to_backup;

  if [ $? -ne 0 ]; then
  echo "Backup of $dir_to_backup failed!"
  exit 1
  fi
}
```

```
for dir in "${DIRECTORIES_TO_BACKUP[@]}"; do
    backup $dir $BACKUP_DIR/$DATE_TAG-$(basename $dir).tar.gz
done

mysqldump -u $DATABASE_USER -p $DATABASE_NAME | gzip > $BACKUP_DIR/$DATE_TAG-$DATABASE

if [ $? -ne 0 ]; then
echo "Backup of database $DATABASE_NAME failed"
exit 1
fi

find $BACKUP_DIR -type f \( -name "*.tar.gz" -o -name "*.sql.gz" \) -mtime +7 -execution

if ["$IS_WEEKLY" = false]; then
find $BACKUP_DIR -type f \( -name "*-week*.tar.gz" -o -name "*-week*.sql.gz" \) -fi

echo "0 0 * * * /path/to/backup_script.sh" |
crontab -
```