CS232 Operating Systems
Assignment 01: Introduction to Bash Scripting

Faraz Ahmed Khan (fk03983)   Syed Ammar Ahmed (sm04050)

Fall 2019

1 Question No. 1

Here you'll write the answer to question 1. You can include any code you have written in the document with the LaTeX listing environments.

Create subsections if a question consists of mulitple parts.

Code directly embedded in LaTeX file.

```
 #!/bin/bash


num=$#

if [ $num == 2 ]
then
    path=$1
    directory=$2
    totaloFiles=0
    totaltxtFiles=0
    if [ -s "$path" ] && [ -d "$directory" ]
    then
        number=$(cut -f 2 -d ',' "$path" | wc -l)
        echo Total Student records: $number

        while IFS=, read -r col1 col2 col3 col4
        do
            echo $totaloFiles
            echo "I␣got:$col2"
            echo $directory/st$col2
            if [ -d "$directory"/st"$col2" ]
            then

                numoFiles=$(find $directory/st$col2 -type f -name '*.o' | wc -l)

                find $directory/st$col2 -type f -name '*.o' -exec rm -rf {} \;

                echo Files deleted 'in' $directory/st$col2: $numoFiles

                totaloFiles=$((totaloFiles + numoFiles))

                txtFiles=$(find $directory/st$col2 -type f -name '*.txt')
                numtextFiles=0

                for file in $txtFiles;

                do
                    firstLine=$(head $file)
                    if [ "$firstLine" == "#/bin/bash" ]
                    then
                        mv "$file" "${file%.*}.sh"
                        numtextFiles=$((numtextFiles + 1))
                    fi
                done
                echo Files renamed 'in' $directory/st$col2: $numtextFiles
                totaltxtFiles=$((totaltxtFiles + numtextFiles))
            else
                echo ERROR: $col3\'s␣directory␣was␣not␣found
␣␣␣␣␣␣␣␣␣␣␣␣fi
␣␣␣␣␣␣␣␣done␣<␣$path
␣␣␣␣else
␣␣␣␣␣␣␣␣echo␣ERROR:␣Either␣path␣or␣directory␣does␣not␣exist.
␣␣␣␣fi
␣␣␣␣echo␣TOTAL␣FILES␣DELETED:␣$totaloFiles
␣␣␣␣echo␣TOTAL␣FILES␣RENAMED:␣$totaltxtFiles
else
␣␣␣␣echo␣ERROR:␣"2␣arguments␣not␣provided"
fi
```

2 Question No. 2

```
#!/bin/bash

arr=("_" "_" "_" "_" "_" "_" "_" "_" "_")


game_run=true

while [ "$game_run" == true ]
do
    read -p "Enter␣your␣move␣(space␣seperated␣integers␣for␣x␣and␣y)" x y
    if [ $x -ge 3 ] || [ $y -ge 3 ]
    then
        echo INVALID MOVE
        continue

    fi
    move_num=$(((3*y)+x))
    free_moves=()
    for value in {0..8}
    do
        cur_ind=${arr[value]}
        if [ "$cur_ind" == "_" ]
        then
            free_moves+=($value)
        fi
    done

    move_allowed=false
    for value in ${free_moves[@]}
    do
        if [ "$value" == "$move_num" ]
        then
            move_allowed=true
        fi
    done


    if [ "$move_allowed" == true ]
    then
        arr[$move_num]="X"
    else
        echo POSITION OCCUPIED
        continue
    fi

    for i in 0 3 6
    do
        start_pos=${arr[$i]}
        pos_1=${arr[$((i+1))]}
        pos_2=${arr[$((i+2))]}
        if [ "$start_pos" != "_" ] && [ "$pos_1" == "$start_pos" ] &&
        [ "$pos_2" == "$start_pos" ]
        then
            echo GAME END
            game_run=false
            break
        fi

    done

    for i in 0 1 2
    do
        start_pos=${arr[$i]}
        if [ "$start_pos" != "_" ] && [ "${arr[$((i+3))]}" == "$start_pos" ] &&
        [ "${arr[$((i+6))]}" == "$start_pos" ]
        then
            echo GAME END
            game_run=false
            break

        fi
    done

    if [ "${arr[0]}" != "_" ] && [ "${arr[4]}" == "${arr[0]}" ] &&
    [ "${arr[8]}" == "${arr[0]}" ]
    then
        echo GAME END
        game_run=false
    fi
```

```
if [ "${arr[2]}" != "_" ] && [ "${arr[4]}" == "${arr[2]}" ] &&
[ "${arr[6]}" == "${arr[2]}" ]
then
    echo GAME END
    game_run=false
fi

if [ "$game_run" == false ]
then
    echo CONGRATULATIONS YOU WON THE GAME
    sleep 1
    break


fi
free_moves=()


for value in {0..8}
do
    cur_ind=${arr[value]}
    if [ "$cur_ind" == "_" ]
    then
        free_moves+=($value)
    fi
done


if [ "${#free_moves[@]}" == 0 ]
then
    echo NO EMPTY POSITION
    break
fi

selectedmove=${free_moves[$RANDOM % ${#free_moves[@]}]}
arr[$selectedmove]="O"


for i in 0 3 6
do
    start_pos=${arr[$i]}
    pos_1=${arr[$((i+1))]}
    pos_2=${arr[$((i+2))]}
    if [ "$start_pos" != "_" ] && [ "$pos_1" == "$start_pos" ] &&
    [ "$pos_2" == "$start_pos" ]
    then
        echo GAME END
        game_run=false
        break
    fi

done

for i in 0 1 2
```

```
do
    start_pos=${arr[$i]}
    if [ "$start_pos" != "_" ] && [ "${arr[$((i+3))]}" == "$start_pos" ]
    && [ "${arr[$((i+6))]}" == "$start_pos" ]
    then
        echo GAME END
        game_run=false
        break

    fi
done

if [ "${arr[0]}" != "_" ] && [ "${arr[4]}" == "${arr[0]}" ] &&
[ "${arr[8]}" == "${arr[0]}" ]
then
    echo GAME END
    game_run=false
fi

if [ "${arr[2]}" != "_" ] && [ "${arr[4]}" == "${arr[2]}" ] &&
[ "${arr[6]}" == "${arr[2]}" ]
then
    echo GAME END
    game_run=false
fi

if [ "$game_run" == false ]
then
    echo YOU LOST THE GAME.
    sleep 1
    break

fi

cols=$( tput cols )
rows=$( tput lines )
input_length=6
half_input_length=$(( $input_length / 2 ))
middle_row=$(( ($rows / 2) - 1 ))
middle_col=$(( (($cols / 2) - $half_input_length)-1 ))

tput clear
tput bold

tput cup $middle_row $middle_col

echo ${arr[0]} ${arr[1]} ${arr[2]}

middle_row=$(( ($rows / 2) ))
middle_col=$(( (($cols / 2) - $half_input_length) - 1 ))

tput cup $middle_row $middle_col

echo ${arr[3]} ${arr[4]} ${arr[5]}
```

```
    middle_row=$(( ($rows / 2) +1 ))
    middle_col=$(( (($cols / 2) - $half_input_length) -1 ))

    tput cup $middle_row $middle_col

    echo ${arr[6]} ${arr[7]} ${arr[8]}
    tput bold

done
```

3 Question No. 3

```
#!/bin/bash

password=$1
if [ ${#password} -le 7 ]; then
        echo Length of password must be greater than equal to 8
        exit
fi

if [[ ! $password =~ [0-9] ]];then
      echo "Password␣does␣not␣contain␣number"
fi
if [[ $password != *['!'#\$%\&*+=-]* ]]
then
  echo "Password␣must␣have␣one␣of␣the␣specified␣special␣characters"
fi
```