**Assignment #2**

Q1)

echo "1. Merging the contents of two files into another"

echo "2. Searching a pattern from a file."

echo "3. Exit from menu "

echo "Enter your choice "

read c

case $c in

1)

echo "Enter the first file name "

read file1

echo "Enter the second file name "

read file2

cat "$file1" "$file2" >merged.txt

echo "The merged file is "

cat merged.txt;;

2)

echo "Enter the file name: "

read file3

echo "Enter the pattern to be searched"

echo "Enter the pattern: "

read pattern

grep "$pattern" "$file3";;

3)

exit;;

\*) echo "invalid option" ;;

esac

op

1. Merging the contents of two files into another

2. Searching a pattern from a file.

3. Exit from menu

Enter your choice

2

Enter the file name:

1.sh

Enter the pattern to be searched

Enter the pattern:

$

case $c in

cat "$file1" "$file2" >merged.txt

grep "$pattern" "$file3";;

Q2)

echo "Enter a word "

read word

if [ $(echo "$word" | rev) = "$word" ];

then

echo "It is a palindrome "

else

echo "It is not a palindrome "

fi

op

Enter a word

casesac

It is a palindrome

Q3)

echo "Enter the file name: "

read f

echo "Enter line number to start from "

read s

echo "Enter line number to stop at "

read e

total=`expr $s + $e - 1`

head -$total $f | tail -$e

op

Enter the file name:

2.sh

Enter line number to start from

2

Enter line number to stop at

3

read word

if [ $(echo "$word" | rev) = "$word" ];

then

Q4)

echo "1. Number of presently active users"

echo "2. Displaying some desired number of lines from top of a file."

echo "3. Updating the access time of a given file to current time."

echo "4. Exit"

echo "Enter your choice "

read c

case $c in

1)

echo "List of users currently logged "

who;;

2)

echo "Enter the file name "

read file

echo "Enter the number of lines to display "

read num

head -$num $file;;

3)

echo "Enter the file name "

read file

touch "$file";;

4)

exit;;

\*) echo "invalid option" ;;

esac

op

1. Number of presently active users

2. Displaying some desired number of lines from top of a file.

3. Updating the access time of a given file to current time.

4. Exit

Enter your choice

2

Enter the file name

3.sh

Enter the number of lines to display

4

echo "Enter the file name: "

read f

echo "Enter line number to start from "

read s

Q5)

echo "Enter size of array "

read n

echo "Enter elements of the array "

i=0

while [ $i -lt $n ]

do

read arr[$i]

let i=i+1

done

echo "Array to be sorted "

echo ${arr[\*]}

for ((i = 0; i < $n; i ++));

do

for ((j = 0; j < $n - i - 1; j ++));

do

if [ "${arr[j]}" -lt "${arr[$((j + 1))]}" ] ;

then

temp=${arr[j]}

arr[$j]=${arr[$((j + 1))]}

arr[$((j + 1))]=$temp

fi

done

done

echo "Descending Bubble Sorted Array "

echo "${arr[\*]}"

op

Enter size of array

4

Enter elements of the array

12

23

34

5

Array to be sorted

12 23 34 5

Descending Bubble Sorted Array

34 23 12 5

Q6)

max=`cut -d ' ' -f1 numbers`

min=`cut -d ' ' -f1 numbers`

for num in `cat numbers`

do

if [ $num -gt $max ]

then

max=$num

fi

done

for num in `cat numbers`

do

if [ $num -lt $min ]

then

min=$num

fi

done

echo "The max elements is $max"

echo "The min element is $min"

op

10 20 30 40 50 60 70 80 90 99 5

The max elements is 99

The min element is 5

Q7)

while :

do

sleep 10

echo "Enjoy your day and press ctrl+c to stop"

done

Q8)

echo "Enter size of array "

read n

echo "Enter elements of the array "

i=0

while [ $i -lt $n ]

do

read arr[$i]

let i=i+1

done

echo "Array to be sorted "

echo ${arr[\*]}

for ((i = 0; i < $n; i ++));

do

for ((j = 0; j < $n - i - 1; j ++));

do

if [ "${arr[j]}" -gt "${arr[$((j + 1))]}" ];

then

temp=${arr[j]}

arr[$j]=${arr[$((j + 1))]}

arr[$((j + 1))]=$temp

fi

done

done

echo "Bubble Sorted Array "

echo "${arr[\*]}"

op

Enter size of array

5

Enter elements of the array

2

4

6

1

7

Array to be sorted

2 4 6 1 7

Bubble Sorted Array

1 2 4 6 7

Q9)

echo "Enter size of array "

read n

echo "Enter elements "

i=0

while [ $i -lt $n ]

do

read arr[$i]

let i=i+1

done

echo "${arr[\*]}"

echo "Enter index number "

read pos

unset arr[$pos]

echo "${arr[\*]}"

op

Enter size of array

4

Enter elements

3

5

2

1

3 5 2 1

Enter index number

3

3 5 2

Q10)

i="y"

echo "Enter name of the file "

read -r file

if [[ ! -e $file ]]; then

touch "$file"

echo "ROLL NUMBER | NAME | CITY" >>"$file"

fi

while [ $i = "y" ];

do

echo "1.Display contents of the file "

echo "2.Display information of the student through roll no "

echo "3.Delete Records "

echo "4.Add Records "

echo "5.Update Records "

echo "6.Exit "

echo "Enter your choice "

read -r ch

case $ch in

1)

sort -k 1 "$file" ;;

2)

echo "Enter Roll No "

read -r i d

grep -i "$id" "$file" ;;

3)

echo "Enter Roll No "

read -r i d

dbs1=$(grep -v "$id" "$file")

echo "$dbs1" >"$file"

echo "Record is deleted"

cat "$file" ;;

4)

echo "Enter Roll No "

read -r id

echo "Enter new student name "

read -r name

echo "Enter city "

read -r city

echo "$id | $name | $city" >>"$file" ;;

5)

echo "Enter Roll No "

read -r id

dbs1=$(grep -v "$id" "$file")

echo "$dbs1" >"$file"

echo "Enter new student name "

read -r name

echo "Enter city "

read -r city

echo "$id | $name | $city" >>"$file" ;;

6)

exit ;;

\*)

echo "Invalid choice " ;;

esac

done

op

Enter name of the file

student

1.Display contents of the file

2.Display information of the student through roll no

3.Delete Records

4.Add Records

5.Update Records

6.Exit

Enter your choice

4

Enter Roll No

2

Enter new student name

Ma

Enter city

kol

1.Display contents of the file

2.Display information of the student through roll no

3.Delete Records

4.Add Records

5.Update Records

6.Exit

Enter your choice

1

2 | Ma | kol

ROLL NUMBER | NAME | CITY

1.Display contents of the file

2.Display information of the student through roll no

3.Delete Records

4.Add Records

5.Update Records

6.Exit

Enter your choice

6

Q11)

s=0

echo "Enter a number :"

read n

t=$n

while [ $n -ne 0 ]

do

r1=$(($n % 10))

rrr=$((r1 \* r1 \* r1))

s=$((s + rrr))

n=$(($n / 10))

done

if [ $t -eq $s ]

then

echo "$t is a ARMSTRONG number"

else

echo "$t is a NON ARMSTRONG number"

fi

op

Enter a number :

153

153 is a ARMSTRONG number

Q12)

echo "Enter 1st number "

read n1

echo "Enter 2nd number "

read n2

echo "Enter 3rd number "

read n3

for i in $n1 $n2 $n3;

do

for j in $n1 $n2 $n3;

do

for k in $n1 $n2 $n3;

do

echo "$i" "$j" "$k"

done

done

done

op

Enter 1st number

6

Enter 2nd number

1

Enter 3rd number

3

6 6 6

6 6 1

6 6 3

6 1 6

6 1 1

6 1 3

6 3 6

6 3 1

6 3 3

1 6 6

1 6 1

1 6 3

1 1 6

1 1 1

1 1 3

1 3 6

1 3 1

1 3 3

3 6 6

3 6 1

3 6 3

3 1 6

3 1 1

3 1 3

3 3 6

3 3 1

3 3 3

Q13)

read -r -p "Enter text: " text

lb=$(echo $text | wc -c)

lb=$(expr $lb - 1)

read -r -p "Enter sub string: " sub

ub=$(echo $sub | wc -c)

ub=$(expr $ub - 1)

n=1

m=1

pos=0

while [ $n -le $lb ]; do

a=$(echo $text | cut -c $n)

b=$(echo $sub | cut -c $m)

if [ $a = $b ]; then

n=$(expr $n + 1)

m=$(expr $m + 1)

pos=$(expr $n - $ub)

r=$(expr $m - 1)

if [ $r -eq $ub ]; then

break

fi

else

pos=0

m=1

n=$(expr $n + 1)

fi

done

echo "Substring Position in String = " $pos

op

Enter text: sobarsh

Enter sub string: bar

Substring Position in String = 3

Q14)

echo "Enter 1st number "

read a

echo "Enter 2nd number "

read b

m=$a

if [ $b -lt $m ]

then

m=$b

fi

while [ $m -ne 0 ]

do

x=$(expr $a % $m)

y=$(expr $b % $m)

if [ $x -eq 0 -a $y -eq 0 ] ; then

echo gcd of $a and $b is $m

break

fi

m=$(expr $m - 1)

done

op

Enter 1st number

32

Enter 2nd number

8

gcd of 32 and 8 is 8

Q15)

echo "Enter the text"

read text

echo

echo "No of words"

echo $text | wc -w

echo

echo "No of characters"

echo $text | wc -m

echo

space=$(expr length "$text" - length `echo "$text" | sed "s/ //g"`)

echo "No of whitespaces $space"

echo

special=$(expr length "${text//[^\~!@#$&\*()]/}")

echo "No of special symbols $special"

echo

op

Enter the text

M$@##$ ^$$^ 533 gygyafd 433%^

No of words

5

No of characters

30

No of whitespaces 4

No of special symbols 7

Q16)

echo "Enter a number :"

read n

f=1

i=2

while [ $i -le $n ]

do

f=$((f \* i))

i=`expr $i + 1`

done

echo "Factorial of $n is : "

echo $f

op

Enter a number :

5

Factorial of 5 is :

120

Q17)

echo "Enter how many "

read n

x=0

y=1

i=2

echo "Fibonacci Series "

echo "$x"

echo "$y"

while [ $i -lt $n ];

do

i=$(expr $i + 1)

z=$(expr $x + $y)

echo "$z"

x=$y

y=$z

done

op

Enter how many

6

Fibonacci Series

0

1

1

2

3

5

Q18)

echo "Enter size of array "

read n

echo "Enter elements of the array "

i=0

while [ $i -lt $n ]

do

read arr[$i]

let i=i+1

done

echo "Array to be sorted "

echo ${arr[\*]}

for ((i = 0; i < $n; i ++));

do

for ((j = 0; j < $n - i - 1; j ++));

do

if [ "${arr[j]}" -lt "${arr[$((j + 1))]}" ] ;

then

temp=${arr[j]}

arr[$j]=${arr[$((j + 1))]}

arr[$((j + 1))]=$temp

fi

done

done

echo "Descending Bubble Sorted Array "

echo "${arr[\*]}"

op

Enter size of array

5

Enter elements of the array

3

6

4

2

1

Array to be sorted

3 6 4 2 1

Descending Bubble Sorted Array

6 4 3 2 1