

BSC SEMESTER 4 (General) PRACTICAL EXAMINATION 2022.

SUBJECT:-COMPUTER SCIENCE

PAPER:-CMMSG -CC4 (PRACTICAL)

REGISTRATION NUMBER:-057-1111-0650-20

ROLL NO:-203057-22-0005

INDEX

SL NO	DATE	ASSIGNMENTS	PAGE NO	TEACHER'S SIGNATURWE
1	19/04/22	Write a shell script to convert the content of a file from lower case to upper case		
2	19/04/22	Write a shell script to count the words, lines and characters of a given file .File name should be provide at runtime		
3	26/04/22	Write a shell script that take a word from user and find out the frequency of the word in a given file.		
4	26/04/22	Write a shell script that gets executed at the moment of user login and displays Good Morning, Good afternoon, Good Evening, Good Night, depending upon the time at which the user log on.		
5	03/05/22	Write a shell script to print Pascal Diamond.		
6	03/05/22	Write a shell script to find a number using sequential search technique.		
7	10/05/22	Write a shell script to find a number using binary search technique.		
8	10/05/22	Write a shell script to sort a set of integer numbers using bubble sort.		
9	17/05/22	Write a shell script to find out the factorial of a given number.		
10	17/05/22	Write a shell script to reverse a string and check whether it is palindrome.		
11	24/05/22	Write a shell script to find the roots of a quadratic equation, considering all possible cases.		
12	24/05/22	Write a shell script for menu based system to insert records for employees with employee ID, Name, Designation, Salary in a data file, also displays records when necessary. Display salary for the employee asked.		

1. Write a shell script to convert the content of a file from lower case to upper case.

Algorithm

Step1: start

step2: enter the file name

step3: read the file

step4: transform the content of the file from lower to upper

step5: stop

Source Code

lower to upper case of word from a given file.

echo "enter the file name"

read f

```
cat $f|tr [:lower:] [:upper:]
```

OUTPUT

```
[gourabdatta@fedora ~]$ sh upper.sh
```

```
enter the file name
```

```
gourab.sh
```

```
HELLO GOURAB
```

2. Write a shell script to count the words, lines and characters of a given file. File name should be provided at runtime.

Algorithm

step1: start

step2: enter the file name.

step3: read the file.

step4: cut the line from the file.

step5: cut the character from the file.

step6: cut the word from the file.

Step7: print the total line.

Step8: print the total characters.

Step9: print the total word.

Step10:Stop

Source Code

counting the word letter and character from a given file

echo "enter the file name"

read f

ln=`cat \$f|wc -l`

ch=`cat \$f|wc -c`

wr=`cat \$f|wc -w`

echo "the number of line is \$ln"

echo "the number of character is \$ch"

echo "the number of word is \$wr"

OUTPUT

[gourabdatta@fedora ~]\$ sh cap1.sh

enter the file name

bs.sh

the number of line is 21

the number of character is 429

the number of word is 84

3. Write a shell script that take a word from user and find out the frequency of the word in a given file.

Algorithm

step1:start

step2:enter the file

step3:read file

step4:cut the word from the file

step5:print the total word .

Step6:stop

Source Code

frequency of word from given file

echo "enter the file name"

read f

wr=`cat \$f|wc -w`

echo "the number of word is \$wr"

OUTPUT

[gourabdatta@fedora ~]\$ sh fow.sh

enter the file name

cap1.sh

the number of word is 48

4. Write a shell script that gets executed at the moment of user login and displays Good Morning, Good afternoon, Good Evening, Good Night, depending upon the time at which the user log on.

Algorithm

Step1:start

Step2:enter the time at which the user log in.

Step3:check the condition whether the input time is 0 to 12.

Step4:show that the desired output for that time.

Step5:check the condition whether the input time is between 12 to 18.

STEP6:-show the desired output for that time.

STEP7:-else show the final output for that time.

Source Code

```
#print good morning,good afternoon,good evening and  
good night according to the user log in time
```

```
hour=$(date +"%H")
```

```
# if it is midnight to mid-afternoon will say G'morning
```

```
if [ $hour -ge 0 -a $hour -lt 12 ]
then
    greet="Good Morning, $USER"
# if it is midafternoon to evening ( before 6 pm) will
say G'noon
elif [ $hour -ge 12 -a $hour -lt 18 ]
then
    greet="Good Afternoon, $USER"
else # it is good evening till midnight
    greet="Good evening, $USER"
fi

14 # display greet
15 echo $greet
```

OUTPUT

```
[gourabdatta@fedora ~]$ sh dateandtime.sh
Good evening, gourabdatta
```

5. Write a shell script to print Pascal Diamond.

Algorithm

Step 1 - Take number of rows to be printed, n.

Step 2 - Make outer iteration I for n times to print rows

Step 3 - Make inner iteration for J to (N - 1)

Step 4 - Print single blank space " "

Step 5 - Close inner loop

Step 6 - Make inner iteration for J to I

Step 7 - Print nCr of I and J

Step 8 - Close inner loop

Step 9 - Print *NEWLINE* character after each inner iteration

Step 10 – Return

Step 11:stop

Source Code

```
# Pascal Diamond
```

```
echo enter limit
```

```
read num

for (( i=1;i<=$num ;i++))
do
for (( j=$num;j>=i;j-- ))
do
echo -n " "
done
for (( c=1;c<=i;c++ ))
do
echo -n " *"
sum=`expr $sum + 1`
done
echo ""
done
d_max=`expr $num - 1`
for (( i=$d_max;i>=1;i--))
do
for (( j=i;j<=$d_max;j++ ))
do
if [ $j -eq $d_max ]
then
echo -n " "
```

```
fi
echo -n " "
done
for (( c=1;c<=i;c++ ))
do
echo -n " *"
sum=`expr $sum + 1`
done
echo ""
done
```

OUTPUT

```
[gourabdatta@fedora ~]$ sh pascaldiamond.sh
```

```
enter limit
```

```
10
```

```

  *
 * *
* * *
* * * *
* * * * *
```

[illegible]

6. Write a shell script to find a number using sequential search technique.

Algorithm

Step 1: Set i to 1

Step 2: if $i > n$ then go to step 7

Step 3: if $A[i] = x$ then go to step 6

Step 4: Set i to $i + 1$

Step 5: Go to Step 2

Step 6: Print Element x Found at index i and go to step 8

Step 7: Print element not found

Step 8: Exit

Source Code

#search a number position using sequential search

echo enter the limit

read n

echo Enter the numbers

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

do

read m

a[i]=\$m

done

for((i=1; i<n; i++))

do

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

do

if [\${a[\$j]} -gt \${a[\$j+1]}]

then

t=\${a[\$j]}

a[\$j]={{a[\$j+1]}}

a[\$j+1]=\$t

fi

done

done

echo Sorted array is

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

do

echo \${a[\$i]}

done

echo Enter the element to be searched:

read s

l=0

c=0

u=\$((n-1))

while [\$l -le \$u]

do

mid=\$(((\$l+\$u)/2))

if [\$s -eq \${a[\$mid]}]

then

c=1

break


```
elif [ $s -lt ${a[$mid]} ]  
then  
u=$(( $mid-1 ))  
else  
l=$(( $mid+1 ))  
fi  
done  
if [ $c -eq 1 ]  
then  
echo Element found at position $(( $mid+1 ))  
else  
echo Element not found  
fi
```

OUTPUT

```
[gourabdatta@fedora ~]$ sh sequential.sh
```

```
enter the limit
```

```
5
```

```
Enter the numbers
```

```
20
```

```
30
```

10

40

2

Sorted array is

2

10

20

30

40

Enter the element to be searched:

2

Element found at position 1

7. Write a shell script to find a number using binary search technique.

Algorithm

Procedure binary_search

Step1:- A ← sorted array

Step2:- $n \leftarrow$ size of array

Step3:- $x \leftarrow$ value to be searched

Step4:- Set lowerBound = 1

Step5:- Set upperBound = n

Step6:-while x not found

Step7:-if upperBound < lowerBound

Step8:- EXIT: x does not exists.

Step9:- set midPoint = lowerBound + (upperBound - lowerBound) / 2

**Step10:- if A[midPoint] < x
set lowerBound = midPoint + 1**

**Step11:- if A[midPoint] > x
set upperBound = midPoint - 1**

Step12:- if A[midPoint] = x

Step13:- EXIT: x found at location midPoint

Step14:- end while

Step15:-end procedure

Source Code

binary search for search a number location in given inputs

```
echo Enter the number of elements:  
read n  
echo Enter the array elements:  
for (( i=1; i<=n; i++ ))  
do  
    read a[$i]  
done  
echo Enter the element to be searched:  
read item  
j=1  
while [ $j -lt $n -a $item -ne ${a[$j]} ]  
do  
    j=`expr $j + 1`  
done  
if [ $item -eq ${a[$j]} ]  
then  
    echo $item is present at location $j  
else  
    echo $item is not present in the array.  
fi
```

OUTPUT

[gourabdatta@fedora ~]\$ sh bs.sh

Enter the number of elements:

4

Enter the array elements:

20

10

16

4

Enter the element to be searched:

4

4 is present at location 4

8. Write a shell script to sort a set of integer numbers using bubble sort.

Algorithm

procedure bubbleSort(list : array of items)

Step1:- loop = list.count;

Step2:- for i = 0 to loop-1 do:

Step3:- swapped = false

Step4:- for j = 0 to loop-1 do:

Step5:- /* compare the adjacent elements */

Step6:- if list[j] > list[j+1] then

Step7:- /* swap them */

Step8:- swap(list[j], list[j+1])

Step9:- swapped = true

Step10:- end if

Step11:- end for

**Step12:- /*if no number was swapped that means
array is sorted now, break the loop.*/**

Step13:- if(not swapped) then

Step14:- break

Step15:- end if

Step16:- end for

Step17:-end procedure return list

Source Code

```
# Sorting the array in Bash  
# using Bubble sort  
#!/bin/bash  
echo "enter maximum number"  
read n  
# taking input from user  
echo "enter Numbers in array:"  
for (( i = 0; i < $n; i++ ))  
do  
read nos[$i]  
done  
#printing the number before sorting  
echo "Numbers in an array are:"  
for (( i = 0; i < $n; i++ ))  
do  
echo ${nos[$i]}  
done  
# Now do the Sorting of numbers  
for (( i = 0; i < $n ; i++ ))  
do  
for (( j = $i; j < $n; j++ ))  
do
```

```
if [ ${nos[$i]} -gt ${nos[$j]} ]; then
t=${nos[$i]}
nos[$i]=${nos[$j]}
nos[$j]=$t
fi
done
done
# Printing the sorted number
echo -e "\nSorted Numbers "
for (( i=0; i < $n; i++ ))
do
echo ${nos[$i]}
done
```

OUTPUT

```
[gourabdatta@fedora ~]$ sh bubblesort.sh
enter maximum number
5
enter Numbers in array:
20
10
```


30

40

50

Numbers in an array are:

20

10

30

40

50

Sorted Numbers

10

20

30

40

50

9. Write a shell script to find out the factorial of a given number.

Algorithm

Step 1: Start

Step 2: Declare Variable n, fact, i

Step 3: Read number from User

Step 4: Initialize Variable fact=1 and i=1

Step 5: Repeat Until $i \leq \text{number}$ 5.1 fact=fact*i 5.2 i=i+1

Step 6: Print fact

Step 7: Stop

Surce Code

```
#factorial of a given number  
echo "enter the numbers of element"  
read n  
fac=1  
for((i=1;i<=$n;i++))  
do  
fac=`expr $fac \* $i`  
done  
echo "the factorial is $fac"
```

OUTPUT

```
[gourabdatta@fedora ~]$ sh factorial2.sh
```

enter the numbers of element

5

the factorial is 120

10. Write a shell script to reverse a string and check whether it is palindrome.

Algorithm

Step1:Split the word into an array, saving it into a variable.

Step2:Reverse the array.

Step3:Put it back together.

Step4:Compare the initial string to the reversed one.

Step5:Stop

Source Code

```
echo Enter the string  
read s  
echo $s>temp  
rvs="$(rev temp)"  
if [ $s = $rvs ]  
then  
echo "it is palindrome"  
else  
echo " it is not a Palindrome"  
fi
```

OUTPUT

```
[gourabdatta@fedora ~]$ sh rsp.sh
```

```
Enter the string
```

```
Gourab
```

```
it is not a Palindrome
```

```
[gourabdatta@fedora ~]$ sh rsp.sh
```

```
Enter the string
```

```
101
```

it is palindrome

11. Write a shell script to find the roots of a quadratic equation, considering all possible cases.

Algorithm

Step 1. Start

Step 2. Read the coefficients of the equation, a, b and c from the user.

Step 3. Calculate discriminant = $(b * b) - (4 * a * c)$

Step 4. If discriminant > 0:

4.1: Calculate root1 = $(-b + \text{sqrt}(\text{discriminant})) / (2 * a)$

4.2: Calculate root2 = $(-b - \text{sqrt}(\text{discriminant})) / (2 * a)$

4.3: Display “Roots are real and different”

4.4: Display root1 and root2

Step 5: Else if discriminant = 0: 5.1: Calculate root1 = $-b / (2 * a)$

5.2: root2 = root1

5.3: Display “Root are real and equal”

5.4: Display root1 and root2

Step 6. Else: 6.1: Calculate real = $-b / (2 * a)$

6.2: Calculate imaginary = $\text{sqrt}(-\text{discriminant}) / (2 * a)$

6.3: Display “Roots are imaginary”

6.4: Display real, “±” , imaginary, “i”

Step 7. Stop

Source Code

Quadratic Equation formula for solving the equation

echo Enter the coefficient of x^2 :

read a

echo Enter the coefficient of x:

read b

echo Enter the constant term:

read c

f=`echo "-(\$b)" |bc`

p=`expr 2 * \$a`

if [\$a -ne 0]

then

d=`echo \(\ (\$b * \$b \) - \(\ 4 * \$a * \$c \) \) | bc`

if [\$d -lt 0]

then

x=`echo "-(\$d)" | bc`

s=`echo "scale=2; sqrt (\$x)" | bc`

echo The first root is:

echo "(\$f + \$s i) / \$p"

echo The second root is:

echo "(\$f - \$s i) / \$p"

elif [\$d -eq 0]

then

res=`expr \$f / \$p`

echo The root is: \$res

else

s=`echo "scale=2; sqrt(\$d)" | bc`

res1=`echo "scale=2; (\$f + \$s) / (\$p)" | bc`

res2=`echo "scale=2; (\$f - \$s) / (\$p)" | bc`

echo The first root is: \$res1

echo The second root is: \$res2

fi

else

echo Coefficient of x^2 can not be 0.

fi

OUTPUT

[gourabdatta@fedora ~]\$ sh quadratic.sh

Enter the coefficient of x^2 :

2

Enter the coefficient of x :

3

Enter the constant term:

4

The first root is:

$(-3 + 4.79 i) / 4$

The second root is:

$(-3 - 4.79 i) / 4$

12 .Write a shell script for menu based system to insert records for employees with employee ID, Name, Designation,Salary in a data file, also displays records when necessary. Display salary for the employee asked.

Algorithm

Step1:-Start

Step2:-Enter the name of the file

Step3:-Enter the employee of the name

Step4:-Read f,n

Step5:-turn a loop from1 to upto the number of employee

Step6:-Enter the name of Employee

Step7:-Enter the Employee id

Step8:-Enter the designation of the employee

Step9:-enter the salary of the employee

Step10:- Read ename,edesig,eid,salary

Step11:-Store the data in the file.

Step12:-enter theemployee name to be searched.

Step13:-Read nm.

Step14:-print the salary of the employee name to be searched.

Step15:-end

INPUT

echo "enter the file name"

read f

echo "enter the number of employee "

read n

for ((i=1;i<=\$n;i++))

do

echo "enter the employee name"

read ename

echo "enter the employee id"

read eid

echo "enter the designation of the employee"

read edesig

echo "enter the salary of employee"

read esalary

echo "\$ename|\$eid|\$edesig|\$esalary">>\$f

done

echo "enter the employee name to be searched"

read nm

echo "The salary of \$nm is `grep \$nm \$f|cut -d '|' -f4`"

OUTPUT

sh progra12.sh

enter the file name

file12

enter the number of employee

3

enter the employee name

Gourab

enter the employee id

g1

enter the designation of the employee

Manager

enter the salary of employee

40000/-

enter the employee name

Jyotiska

enter the employee id

j2

enter the designation of the employee

Ceo

enter the salary of employee

100000/-

enter the employee name

Argho

enter the employee id

a3

enter the designation of the employee

Storekeeper

enter the salary of employee

20000/-

enter the employee name to be searched

Gourab

The salary of Gourab is 40000/-

