



Estd. Yr. 2000

# ARYA College of Engineering & I.T.

www.aryacollege.in



Estd. Yr. 2000

# ARYA College of Engineering & I.T.

www.aryacollege.in

## CONTENTS & EVALUATION REPORT

Code... 9.C.P.S.Y - 2.Y

Name of the Laboratory.....L.S.P.....

Code... 9.C.D.S.Y - 2.Y

Name of the Laboratory.....L.S.P.....

To be filled by the student

To be filled by the faculty

S.No.	Name of The Experiment	Page No.	To be filled by the student			To be filled by the faculty		
			Date of Performance	Date of Evaluation	Performance	Record	Viva	Total
1	use of basic shell commands ls -mdu -c cat /etc							
2	command related to i/o redirection and piping, process control commands							
3	shell script based on control structure if - then - else if - then - else if nested if							
4	shell programming use of loops until, for loop shell							
5	looping code							
6	shell programming function							
7	shell script to print different steps diamond triangle hollow square etc							
8	*Shell Programming							

Programing  
for  
string code  
structure  
use of break



Estd. Yr. 2000

ARYA 1<sup>ST</sup> OLD CAMPUS

www.aryacollege.in

# ARYA College of Engineering & I.T.

Kukas, Jaipur

Experiment No. :

Date :

Q1

Write the use of basic unix shell commands ls, mkdir, rmkdir, cd, cat, banner, touch, file, wc, sort, cut, grep, dd, dfspace, du, ulimit

(1) LS This command lists the

content of the current directory

ls <directory name> with location

To list the content of directory

/home/user, we write ls /home/user

(2) Cd This command changes

current directory

cd <directory name> with location

To change the current directory

to /home/user we write cd /home/user

(3) Mkdir This command is used to

create a new directory at

mkdir <directory name>

To create new directory 'test' we

write mkdir test



Experiment No. :

Date :

(9) `rmdir [remove directory]` This command removes an empty directory

Syntax `rmdir < directory name >`

example To remove directory name 'test' we write command `rmdir test`

(5) `cat [concatenate and display]` this command concatenates files & print them to standard output.

Syntax `cat < file name >`

example To display content of 'file.txt' we write command `cat file.txt`

(6) `Banner` This command is used to create large ASCII text banners

Syntax `banner text`

example To create banner for word 'Hello'  
`banner Hello`

(7) `Touch` This command is used to create an empty file

Syntax `touch < file name >`

example To create a file name 'file.txt' we write `touch file.txt`



Estd. Yr. 2000

ARYA 1<sup>ST</sup> OLD CAMPUS

www.aryacollege.in

# ARYA College of Engineering & I.T.

Kukas, Jaipur

Experiment No. :

Date :

(8) File This command determines the type of a file

Syntax file [options] file-name

example To determine the type of a file named 'file.txt' we write # file file.txt

(9) WC [word count] counts number of lines, words & character in a file

Syntax WC < file name >

example To count number of line words, and bytes in a file named "example.txt" we write WC example.txt

(10) Sort This command sorts the line of a file

Syntax sort [options] file-name

example Sort lines of a file named 'example.txt' alphabetically we write sort example.txt

(11) Cut this command extracts cut fields from file and extracts desired section from file



Experiment No. :

Date :

Syntax `cut -d (delimiter) -f (column no) < file name`  
example To extract first column from a file named "example.txt" using a delimiter ";" (comma) we write `cut -d ';' -f 1 example.txt`

(12) Grep [Global regular expression print] This command searches for pattern in a file

Syntax `grep [option] 'pattern' file name`  
example search for word "example" in a file name "example.txt" we write `grep 'example' example.txt`

(13) dd [data duplicator] This command is used to convert and copy file's data

Syntax `dd if=<input file> of=<output file>`  
Output To make backup of sda drive we write command

`dd if=/dev/sda of=backup.img`  
`Conv=noerror, sync bs=1M`



Estd. Yr. 2000

ARYA 1<sup>ST</sup> OLD CAMPUS

www.aryacollege.in

# ARYA College of Engineering & I.T.

Kukas, Jaipur

Experiment No. :

Date :

(14)

df (disk free) This command displays the free disk space on the system

syntax

df [option] [file]

output

To display disk space usage for all mounted file system we write df -h

(15)

du (disk usage) This command displays amount of disk space used by files and directories

syntax

du [option] [director name]

output

To display disk usage of /home/user example we write command du -h /home/user

(16)

ulimit This command is used to set and display user resource limit

syntax

ulimit [options]

output

To display all limits of current user we write command

example

# ulimit -a



Q 2 Command related to inode, I/O redirection and piping, process control commands, mails

\* command related to inode An inode is a file data structure that stores information about any Linux file except its name and data information contained in an inode

- (1) File size
- (2) Device on which the file is stored
- (3) User and group IDs associated with file
- (4) Permissions needed to access the file

(1) ls -i This command is used to get inode number of a file

Syntax ls -i [file or directory]

Example To get inode number of directory /etc/passwd we use command ls -i /etc/passwd

(2) Stat Display detailed information about a file including inode number

Syntax stat <file name>



Estd. Yr. 2000  
**ARYA** College of Engineering & I.T.

Kukas, Jaipur

Experiment No.:

Date:

Example To get details and inode numbers of file

example `tat` see write command

stat example, tat

(3) df -i This command displays inode usage information for a mounted filesystem

Syntax `df -i [directory]`

Example To get inode information of /dev/sda2 we write command `df -i /dev/sda2`

\* Commands related to IO redirection

i) Output redirection [ '>' ] This operator redirect standard output of a command to a file, creating or overwriting it

Syntax Command  $\rightarrow$  output file

Example To redirect output of find command in a file `find /etc/ -name passwd > output.txt` not output. but

(ii) Append output redirection [ '>>' ] This operator redirect standard output of



Estd. Yr. 2000

# ARYA College of Engineering & I.T.

Kukas, Jaipur

Date:

## OUTPUT

```
/etc / password  
/etc / shadow / password  
/etc / shadow
```

To open file in append mode

(ii) command to a file by appending it with previous data of file  
Syntax command >> output\_file  
example To append data in output file. txt  
command  
`find /etc / - name shadow >> output.txt`

(iii) Input redirection [`<`] This operator  
Redirects standard input from a file to command  
Syntax command < [input\_file]  
example redirect input from data.txt to sort command  
`sort < data.txt`

## OUTPUT

(iv) Error redirection [`'2>'`] It's operator  
redirects standard error of a command to a file

Syntax [command] 2> error\_file  
example To redirect error of cat command  
`cat not existing file 2> error.txt`  
`cat error.txt`



Estd. Yr. 2000  
**ARYA** College of Engineering & I.T.

ARYA 1<sup>ST</sup> OLD CAMPUS  
Kukas, Jaipur

[www.aryacollege.in](http://www.aryacollege.in)

## OUTPUT

cat : not existin - file : No such file or directory

lsch : cd : not - existin' dir : No Such file or directory

(v) Append error redirection [ '2>>' ] This operator redirect and offend standard error of a command to a file  
 [ Command ] 2>> error - file  
 To redirect and offend error in previous file  
 cd not - existin - dir 2>> err. txt  
 cat err .txt

(vi) Redirecting error and output to some file [ '>>' ] This operator redirects standard error and output in same file  
 [ Command ] &> output - file  
 Syntax to redirect error and output of ls command  
 ls - l / nonexistin' directory &> err. txt

(vii) Pipe operator [ ' | ' ] This operator sends the output of one command as input to another command  
 Syntax command 1 | command 2  
 example to do both redirection and displaying output of a command using '|';

ls - l | tee output. txt

## OUTPUT

[1] + stopped tail - F /etc/passwd

[1] + tail -F /etc/passwd &

- (i) log log command resumes the suspended process while keep them running in background

log [1].job7

Example To move tail command to background press tail -F /etc/passwd

ctrl + Z

jobs

log %.1

[1] + Running tail -F /etc/passwd

(ii) fg fg command moves a background process or job to current shell environment to foreground

Symbol

fg [1].job7

example To move tail command back to foreground

fg %.1

## OUTPUT

root 2756 0.0 0.0 220988 pts/6 tail -F /etc

foreground

Symbol

fg [option]

example

To display information about 'tail' process

he - own | grep 'tail'



Estd. Yr. 2000

ARYA College of Engineering & I.T.

www.aryacollege.in

Experiment No.:

Date:

Kukas, Jaipur

	Process control commands



ESTD. YR. 2000  
ARYA 1<sup>ST</sup> OLD CAMPUS

Kukas, Jaipur  
ARYA College of Engineering & I.T.

[www.aryacollege.in](http://www.aryacollege.in)

## OUTPUT

[1]+ killed tail - F /etc/hostname

tail - F /etc/hostname

(i) kill This command sends signal to process, allowing user to terminate or control behaviour of process  
Syntax kill [option] < pid >  
example to terminate process tail - F we write command  
kill - 9 2756

\* Command related to mail

(1) Sending an email use command  
mail using 8smtp

echo "Message body" | smtph recipient@

email.com

(2) Receiving or reading message body from a file through Smtp and input redirection we can read mail

smtph recipient@mail.com < message.txt

Experiment No.:

Date:



Estd. Yr. 2000

ARYA 1<sup>ST</sup> OLD CAMPUS  
Kukas, Jaipur

www.aryacollege.in

Experiment No.: 3

Date:

### OUTPUT 3.1

```
Enter first value 5
Enter second value 6
Enter third value 7
```

7 is greatest among three numbers

```
#!/bin/bash
read -p "Enter first value" a
read -p "Enter second value" b
read -p "Enter third value" c
if [ $a -gt $b ] && [ $a -gt $c ]
then
    echo "$a is greatest among three
numbers"
else
    echo "$b is greatest among
three numbers"
fi
```

Page No. ....



Estd. Yr. 2000

# ARYA College of Engineering & I.T.

Experiment No. :

Kukas, Jaipur

Date:

OUTPUT 3.2

Enter a year : 2016

2016 is a leap year

3.2 To find a year is a leap year or not

#!/bin/bash

read - p "Enter a year : " year

```
if [ $(($((year \% 4)) - 2)) -eq 0 ] &&  
[ $(($((year \% 100)) - ne 0)) -eq 0 ] ||  
[ $(($((year \% 400)) - eq 0 )]
```

then

echo "\$ year is a leap year."

else

echo "\$ year is not a leap year"

fi



Estd. Yr. 2000

# ARVA College of Engineering & I.T.

Kukas, Jalandhar

Date:

Experiment No.:

Page No. ....

OUTPUT 3.3

Enter first angle of triangle 40

Enter second angle of triangle 50

Enter third angle of triangle 90

It is a valid triangle

```
3.3 To input angle of a triangle to  
to find out whether it is a valid triangle  
#1 / bin / bash  
read -p "Enter first angle of the  
triangle" a  
read -p "Enter second angle of the  
triangle" b  
read -p "Enter third angle of the  
triangle" c  
if [ $((a+b+c)) -eq 180 ]  
then  
    echo "It is a valid triangle"  
else  
    echo "It is not a valid triangle"  
fi
```



Experiment No. :

Date :

3.9

To check whether a character is alphabet digit or special character

# ! / \ \$ ^ / %

OUTPUT 3.9

Enter any character : A  
It is an alphabet

```
read -p "Enter any character " ch
if [[ $ch == [A-Z] ]] || [[ $ch == [a-z] ]]
then
    echo "It is an alphabet"
elif [[ $ch == [0-9] ]]
then
    echo "It is a digit"
else
    echo "It is a special character"
fi
```



Estd. Yr. 2000

**ARYA COLLEGE OF ENGINEERING & I.T.**

ARYA 1<sup>ST</sup> OLD CAMPUS  
Kukas, Jaipur

www.aryacollege.in

Experiment No.:

Date:

3.5 To calculate Profit or Loss

#!/bin/bash

```
read -p "Enter selling price" $p
read -p "Enter cost price" $c
if [ $sp -gt $cp ]
then
    profit=$((($sp - $cp)))
    echo "There is a profit of"
    echo "$profit"
else
    if [ $cp -gt $sp ]
    then
        loss=$((($cp - $sp)))
        echo "There is a loss of"
        echo "$loss"
    else
        echo "There is neither profit nor loss"
    fi
fi
```



Experiment No.:

Date:

### O 4 Shell Programming - Looking while, until, for loop

O 4.1 Write a shell script to print all even and odd numbers from 1 to 10

Ans

```
#!/bin/bash
```

```
echo "Even numbers"
for (( i=2; i<=10; i+=2 ))
do
```

```
    echo "$i"
```

done

### Odd numbers

```
1
3
5
7
9
echo "odd numbers"
for (( i=1; i<=10; i+=2 ))
do
    echo "$i"
```

done

## OUTPUT 4.2

Enter any number: 5

5	X	1	=	5
5	X	2	=	10
5	X	3	=	15
5	X	4	=	20
5	X	5	=	25
5	X	6	=	30
5	X	7	=	35
5	X	8	=	40
5	X	9	=	45
5	X	10	=	50



Q 4.2 Write a script to print table of given number.

```
#!/bin/bash
read -p "Enter any number: " n
for (( i=1 ; i<=10 ; i++ ))
do
  echo "$n X $i = $(($i*n))"
done
```

## OUTPUT 4.3

Q 4.3

Write a shell script to calculate factorial of a given number

```
Ans #!/bin/bash
read -p "Enter any number: " n
```

ans=1

until [ \$n -eq 1 ]

do

ans=\$(( \$ans \* \$n ))

n=\$(( \$n - 1 ))

done

Echo "Factorial of Given number is \$ans"

OUTPUT 4.4

Sum of the even number till 10 is 30

OUTPUT 4.5

Enter any number 125

Sum of digit of given  
Number is 8

Q4.4 Script to print sum of all numbers from 1 to 10

#!/bin/bash  
Counter=10  
Ans=0

until [ \$Counter -lt 1 ]  
do

if [ \$((\$Counter % 2)) -eq 0 ]  
then

Ans=\$((\$Ans + Counter))  
fi  
Counter=\$((\$Counter - 1))

done

Echo " Sum of Even numbers till 10 is  
\$Ans "

Q4.5 Write a script to print sum of digit of any number  
#!/bin/bash  
read -p " Enter any number " num  
Ans=0  
while [ \${num%?} -ne 0 ]  
do  
temp=\$((num%10))  
Ans=\$((Ans+temp))  
done  
Num=\$((num/10))

echo " Sum of the digit of given number is \$Ans "  
Page No. ....

# OUTPUT S. 1

Input two values

Enter value of a : 10

Enter value of b : 10

Enter your choice of operation you

want to perform

1 is for addition

2 is for subtraction

3 is for multiplication

4 is for division

Sum is 20

Experiment No.: 5

Date:

Shell programming - case structure, use of lot  
Write a shell script program to make a basic calculator which perform addition, subtraction, Multiplication, division

```
#!/bin/bash
```

```
echo "Enter two values"
```

```
read -p "Enter value of a" a
```

```
read -p "Enter value of b" b
```

```
echo "1 is for addition"
```

```
echo "2 is for subtraction"
```

```
echo "3 is for multiplication"
```

```
echo "4 is for division"
```

```
read ch
```

```
case $ch in
```

```
1) Sum=$((a+b))
```

```
echo "Sum is $sum"
```

```
2) Sub=$((a-b))
```

```
echo "Subtraction is $sub"
```

```
3) mul=$((a*b))
```

```
echo "Multiplication is $mul"
```

```
4) div=$((a/b))
```

```
echo "Division is $div"
```



Estd. Yr. 2000  
**ARYA** College of Engineering & T.I.

ARYA 1<sup>st</sup> OLD CAMPUS  
www.aryacollege.in  
Kukas, Jaipur

Experiment No.:

Date:

Output 5.2

Enter the number to get corresponding day of week: 1

Monday

Q5.2

\* ) echo "Not a valid choice"

#!/bin/bash

Read -p "Enter the number to get corresponding day of the week"

case \$ch in

- 1) echo "Monday"
- 2) echo "Tuesday"
- 3) echo "Wednesday"
- 4) echo "Thursday"
- 5) echo "Friday"
- 6) echo "Saturday"
- 7) echo "Sunday"

\* ) echo "Enter a valid choice"

Output

Q5.3

wrote a shell script to print  
starting 4 months having 31 days



Estd. Yr. 2000

ARYA 1<sup>ST</sup> OLD CAMPUS

www.aryacollege.in

Kukas, Jaipur

Experiment No.:

Date:

OUT PUT 5.3

Dr. B.B. Dutt

# !/bin/bash

Count=0

for i in {1..12} do

case \$i in

- 1) echo "January" ((Count++))
- 3) echo "March" ((Count++))
- 5) echo "May" ((Count++))
- 7) echo "July" ((Count++))

done

if [ \$Count -eq 4 ]

then

break

fi

done



Estd. Yr. 2000

**ARYA COLLEGE OF ENGINEERING & I.T.**

Experiment No. :

Date:

ARYA 1<sup>ST</sup> OLD CAMPUS  
www.aryacollege.in

OUTPUT 6.1

Enter a number : 153  
Given number is Armstrong

6  
Q6.1 Write a shell script to find a number is Armstrong or not.

```
#!/bin/bash
function armstrong {
local org=$1
local sum=0
local temp=$org
while [ $temp -ne 0 ] do
local digit=$(( $temp % 10 ))
sum=$(( ($sum + digit * digit * digit) ))
temp=$(( $temp / 10 ))
done
if [ $sum -eq $org ]
then
echo "Given number is Armstrong"
else
echo "Given number is not Armstrong"
fi
}
read -p "Enter a number : " num
armstrong $num
```



Estd. Yr. 2000

**ARYA COLLEGE OF ENGINEERING & I.T.**

Experiment No.:

Date:

www.aryacollege.in  
Kukas, Jaipur

OUTPUT 6.2

Enter a number : 121  
 Given number is palindrome

```
#!/bin/bash
function palindrome {
local org=$1
local temp=$org
local sum=0
while [ $temp -ne 0 ]
do
  local digit=$(( temp % 10 ))
  sum=$(( digit + sum * 10 ))
  temp=$(( temp / 10 ))
done
if [ $sum -eq $org ]
then
  echo "Given number is palindrome"
else
  echo "Given number is not palindrome"
fi
}

Read - p "Enter any number" num
palindrome $num
```

## OUTPUT 6.3

Enter the range 5

Elsonage left S terms

local  $\alpha = 0$

asking "Fibonacci up to \$n\$ to  
for ( $c$  i=0;  $i \leq n$ ;  $i++$ )  
do

```
Echo " $a "
```

$$\alpha = \frac{1}{2} \ln \left( \frac{1 + \sqrt{1 - 4\beta}}{1 - \sqrt{1 - 4\beta}} \right)$$

read - f "Enter the range" num  
l: from: \$num

fascicolo 1

Q6.4 Write a Shell script to find prime numbers.

function  $f$  from  $\mathbb{N}$  to  $\mathbb{N}$  such that  $f(n) = f(m)$  if and only if  $n = m$ .

local temp = 0  
for (( i = 1; i < n; i++ ))



Estd. Yr. 2000

ARYA 1<sup>ST</sup> OLD CAMPUS

www.aryacollege.in

Experiment No.:

Date:

OUTPUT 6.4

Enter any number: 17

Given number is prime number

do

```
if [ $CCn -eq 0 ]
```

then

```
temp = $CCtemp +1 )
```

```
done
```

```
fi
```

```
[ $temp -eq 2 ]
```

then

else

```
echo "Given number is prime number"
```

```
else
echo "Given number is not prime number"
```

```
fi
fi
"Enter any number" num
```

```
primeNumbers $ num
```

Q6.5 Write a shell script to convert binary to decimal and decimal to binary

```
#!/bin/bash
```

function decimal\_to\_binary {  
local val=\$1  
local binary=""



Estd. Yr. 2000  
**ARYA** College of Engineering & T.T.

ARYA 1<sup>st</sup> OLD CAMPUS  
www.aryacollege.in  
Kukas, Jaipur

Experiment No. :

Date :

OUTPUT 6.5

Enter any decimal numbers : 5

Given number in binary form will  
be 101

3

Echo " Given number in binary form  
will be \$binary"

done

Enter any binary number: 101  
Given number in decimal form will  
be 5

function binary -> decimal \$  
local \$val = \$1  
local \$decimal = 0  
local \$base = 1  
local \$temp = 0  
while [ \$val -ne 0 ] do

\$temp = \$ ( (\$val % 10) ) do  
\$decimal = \$ ( \$decimal + \$temp \* \$base )  
\$val = \$ ( \$val / 10 )  
\$base = \$ ( \$base \* 2 )

done  
echo " Given number in decimal form  
will be \$decimal "



## OUTPUT

Enter the number of rows for diamond : 3

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

7. Write a shell script to print different shaped diamond

```
#!/bin/bash
```

Echo "Enter number of rows for the diamond"

read n

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

```
for((j=1; j<=n; j++)) do
```

```
echo -n "
```

done

```
for ((j=1; j<=(2*i-1); j++)) do
```

```
echo -n "*"
```

done

echo

done

```
for ((i=n-1; i>=1; i--)) do
```

```
for ((j=1; j<=i; j--)) do
```

```
echo -n "
```

done

done

## OUTPUT

Enter the number of rows for  
the triangle 4

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

## OUTPUT

Enter length of side for square 4

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



```
for ((i=1; i<= (2*n)-1; i++)) do
    echo -n " "
done
```

Triangle

```
read - p " Enter rows for a triangle "
for ((i=1; i<= n; i++)) do
    for ((j=1; j<= i; j++)) do
        echo -n "* "
    done
```

Echo

done

(c)

Square

```
#!/bin/bash
read -p "Enter the length of side for square "
n
```

```
for ((i=1; i<= n; i++)) do
    for ((j=1; j<= n; j++)) do
        echo -n "* "
    done
```

done

done





Estd. Yr. 2000

# ARYA COLLEGE OF ENGINEERING & TECHNOLOGY

Kukas, Jaipur

Page No. ....

Experiment No.:

Date:

OUTPUT 8.1

Enter number of elements: 5  
Enter 5 elements

1

2

3

4

5

The elements of array are :

1 2 3 4 5

Q 8.1 Write a C program read and print elements of array  
#include <stdio.h>  
int main()  
{  
 int n;  
 printf ("Enter number of elements : ");  
 scanf ("%d", &n);  
 int arr[n];  
 printf ("Enter %d elements : ", n);  
 for (int i = 0; i < n; i++) {  
 scanf ("%d", &arr[i]);  
 }  
 printf ("The elements of array are : ");  
 for (int i = 0; i < n; i++) {  
 printf ("%d", arr[i]);  
 }  
 return 0;  
}

Q8 - 0 Program 8-1 program 8-1.c  
./Program 8-1



## OUTPUT 8.1

```
Enter number of elements 5
Enter the elements of array
1
2
3
4
5
Sum of the elements of
array is 15
```

E. 1 Write a C program to find sum of array of length n.

```
#include <stdio.h>
int main()
{
    int n;
    printf ("Enter number of elements ");
    scanf ("%d", &n);
    int arr[n];
    int sum = 0;
    printf ("Enter the elements of array: ");
    for (int i=0 ; i<n ; i++)
    {
        scanf ("%d", &arr[i]);
        sum = sum + arr[i];
    }
    printf ("Sum of the elements of the
            array is %d", sum);
    return 0;
}
```

Q.C - 0 Program 8-2 Program 8-2



Estd. Yr. 2000

ARYA 1<sup>ST</sup> OLD CAMPUS  
www.aryacollege.in

Experiment No.:

Date:

Output 8-3

Enter number of elements: 5.

Enter 5 elements

5  
4  
3  
2  
1

Reversed array 1 2 3 4 5 -

8.3 Shell programming - Arrays to reverse of an array  
(sim program 8-3.c)

```
#include < stdio.h >

void reversearray (int arr[], int size) {
    int revarr [size];
    for (int i = 0; i < size; i++) {
        revarr [size-i-1] = arr [size-i-1];
    }
    printf ("Reversed array : ");
    for (int i = 0; i < size; i++) {
        printf ("%d", revarr[i]);
    }
}

int main () {
    int n;
    printf ("Enter the number of elmts: ");
    scanf ("%d", &n);
    int arr[n];
    printf ("Enter %d elements: \n", n);
    for (int i = 0; i < n; i++) {
        scanf ("%d", &arr[i]);
    }
    reversearray (arr, n);
    return 0;
}
```

8-3 Program 8-3 Program 8-3.c  
./program 8-3



Estd. Yr. 2000

# ARYA College of Engineering & I.T.

Experiment No.:

Date:

Kukas, Jaipur

OUTPUT 8.4

8.4 write a C program to search an element in an array  
 in an array 8-4. c  
 (a C program 8-4. c  
 #include < stdio.h >

8.4

Enter number of elements : 5  
 Enter elements of array

10  
 5  
 20  
 21  
 25

Enter item to be searched 25  
 25 item found at index 4

```
int main() {
    int item, n, flag;
    flag = 0;
    printf ("Enter number of elements (n)");
    scanf ("%d", &n);
    int a[n];
    for (int i=0; i<n; i++) {
        scanf ("%d", &a[i]);
    }
    printf ("Enter the item to be searched");
    scanf ("%d", &item);
    for (int i=0; i<n; i++) {
        if (a[i] == item) {
            flag = 1;
        }
    }
    if (flag == 1) {
        printf ("\n %d item found at index %d, item, flag);
```



Estd. Yr. 2000

ARYA 1<sup>ST</sup> OLD CAMPUS

www.aryacollege.in

# ARYA College of Engineering & I.T.

Kukas, Jaipur

Experiment No. :

Date :

```
else {  
    printf ("Item not found");  
}  
return 0; }
```

Q8.5 Write a C program to sort array element in ascending or descending order

C program 8-5.c

```
#include <stdio.h>
```

```
void Ascending (int arr[], int n)
```

```
{ int temp;
```

```
for (int i = 0; i < n - 1; i++) {
```

```
    for (int j = i + 1; j < n; j++) {
```

```
        temp = arr[i];
```

```
        arr[i] = arr[j];
```

```
        arr[j] = temp;
```

```
} }
```

```
} }
```

```
void Descending (int arr[], int n)  
{ int temp;
```



Estd. Yr. 2000

# ARYA COLLEGE OF ENGINEERING & I.T.

Experiment No.:

Kukas, Jaipur

Date:

OUTPUT 8.5

Enter number of elements 5

Enter 5 elements

10

5

8

11

80

Array in ascending order :

5 8 10 11 80

Array in descending order :

80 11 10 8 5

```

for (int j = 0; j < n - 1; j++) {
    for (int i = j + 1; i < n; i++) {
        if (arr[i] < arr[j]) {
            temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
}

void printarray(int arr[], int n) {
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
}

int main() {
    int n;
    printf("Enter number of elements ");
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    Ascending (arr, n);
    printf("Array in ascending order ");
    printarray(arr, n);
    Descending (arr, n);
    printf("Array in descending order ");
    printarray(arr, n);
}

```

Return 0;