PROGRAM TO DISPLAY STUDENT DETAILS

```
echo "Enter your name";
read name;
echo "Enter your semester";
read sem;
echo "Enter your batch";
read batch;
echo "Enter your roll number";
read rollno;
echo "Enter your register number";
read regno;
 echo "Name: $name"
 echo "Semester: $sem"
 echo "Batch: $batch"
 echo "Roll Number: $rollno"
 echo "Register Number: $regno"
```

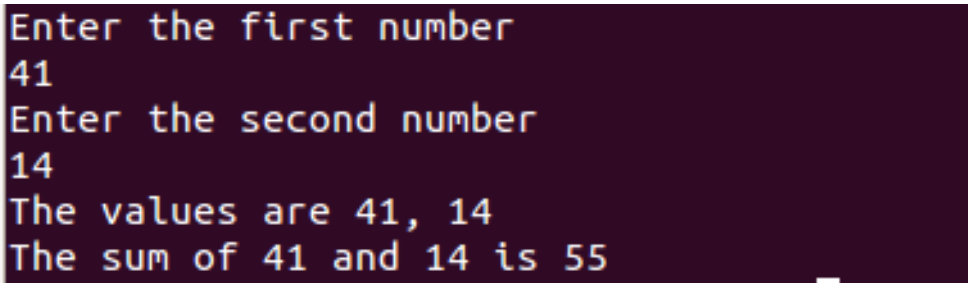
OUTPUT

```
Enter your name
Aromal
Enter your semester
4
Enter your batch
B
Enter your roll number
15
Enter your register number
22CS032
Name: Aromal
Semester: 4
Batch: B
Roll Number: 15
Register Number: 22CS032
```

PROGRAM TO DISPLAY SUM OF TWO NUMBERS

```
echo "Enter the first number";
read a;
echo "Enter the second number";
read b;
echo "The values are $a, $b";
sum=$(($a+$b));
echo "The sum of $a and $b is $sum"
```

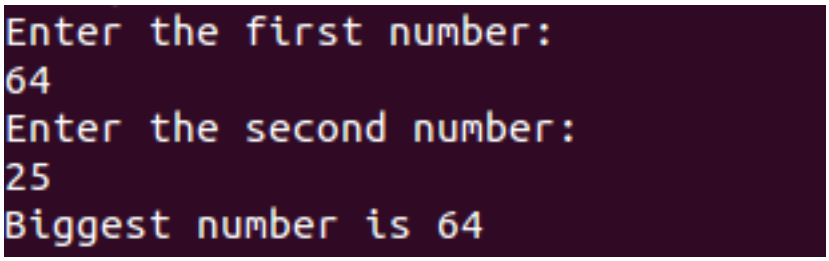
OUTPUT



PROGRAM TO DISPLAY LARGEST OF TWO NUMBERS

```
echo "Enter the first number:";
read x;
echo "Enter the second number:";
read y;
if(($x>$y))
then
echo "Biggest number is $x";
else
echo "Biggest number is $y";
fi
```

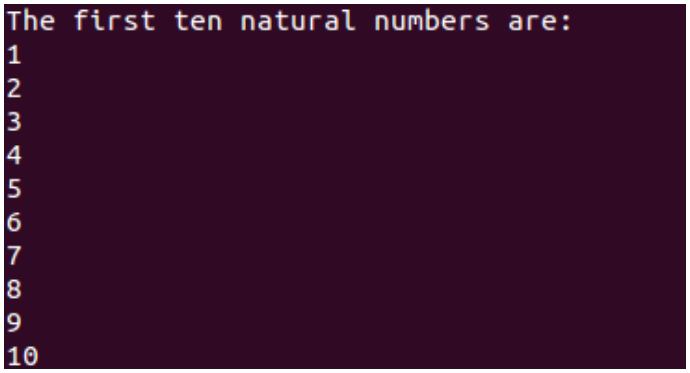
OUTPUT



PROGRAM TO PRINT FIRST 10 NATURAL NUMBERS

```
echo "The first ten natural numbers are:"
for((i=1;i<=10;i++))
do
echo $i
done
```

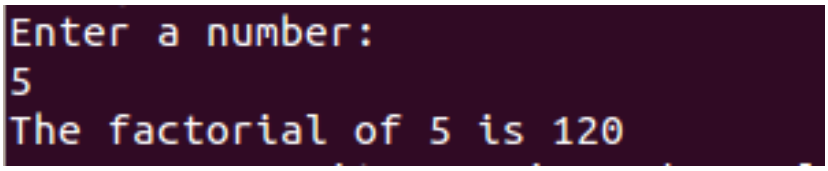
OUTPUT



PROGRAM TO FACTORIAL OF A NUMBER

```
echo "Enter a number: "
read num
fact=1
for(( i=1; i<=num; i++ ))
do
  fact=$[ $fact * $i ]
done
echo "The factorial of $num is $fact"
```

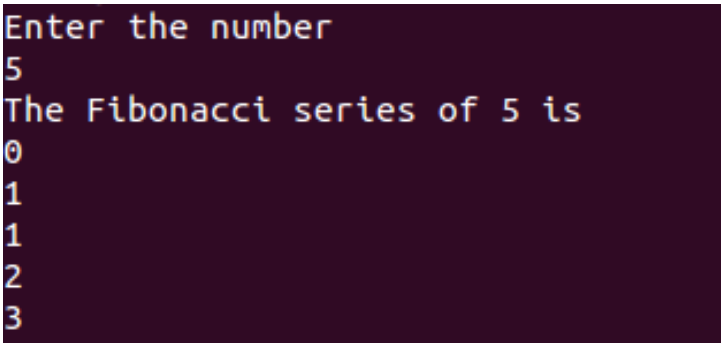
OUTPUT



PROGRAM TO FIND FIBONACCI SERIES

```
echo "Enter the number"
read n
x=0
y=1
echo "The Fibonacci series of $n is"
for((i=1;i<=n;i++))
do
    echo "$x"
    f=$(($x + $y))
   x=$y
   y=$f
done
```

OUTPUT



PROGRAM TO IMPLEMENT SIMPLE CALCULATOR

```
echo "Enter the first number"
read n1
echo "Enter the second number"
read n2
echo "1.Addition"
echo "2.Subtraction"
echo "3.Multiplication"
echo "4.Division"
echo "Choose the operation(1-4)"
read op
case $op in
1)
   rs=$(($n1 + $n2))
   echo "The sum is $rs";;
2)
   rs=$(($n1 - $n2))
   echo "The difference is $rs";;
3)
   rs=$(($n1 * $n2))
   echo "The product is $rs";;
4)
   rs=$(($n1 / $n2))
   echo "The quotient is $rs";;
*)
   echo "Wrong choice entered";;
esac
```

OUTPUT

```
Enter the first number
5
Enter the second number
6
1.Addition
2.Subtraction
3.Multiplication
4.Division
Choose the operation(1-4)
3
The product is 30
```

Program to implement fork(), getpid(),getppid(),wait(), exit()

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <stdlib.h>
int main()
{
pid_t childpid=fork();
if(childpid==-1)
{
printf("Child Creation Unsuccessfull");
}
else if(childpid==0)
{
printf("Child process");
printf("\nPID: %d", getpid());
printf("\nParent PID: %d", getppid());
}

else
{
printf("Parent Process\n");
printf("PID: %d\n", getpid());
printf("Child PID: %d\n",childpid);
wait(NULL);
printf("\nChild Finished");
exit(0);
}

return 0;
}
```

OUTPUT

```
Parent Process
PID: 4293
Child PID: 4294
Child process
PID: 4294
Parent PID: 4293
Child Finishedcseb1@sjcet-H81M-DS2:
```

Program to implement execvp()

first.c

```
#include <stdio.h>
#include <unistd.h>
int main()
{
printf("I am program first.c called by second.c\n");
printf("Bye\n");
return 0;
}
```

second.c

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main()
{
char *args[]={"./EXEC",NULL};
execvp(args[0],args);
return 0;
}
```

OUTPUT

```
I am program first.c called by second.c
Bye
```

Program to implement stat()

```
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<stdlib.h>
void main()
{
char *path, path1[10];
struct stat *nfile;
nfile=(struct stat *) malloc(sizeof(struct stat));
printf("Enter Filename:");
scanf("%s",path1);
stat(path1,nfile);
printf("User Id: %d\n",nfile->st_uid);
printf("Blocksize:%ld\n",nfile->st_blksize);
printf("Last access time:%ld\n",nfile->st_atime);
printf("Last modification:%ld\n",nfile->st_mtime);
printf("Production mode:%d\n",nfile->st_mode);
printf("Size of file:%ld\n",nfile->st_size);
printf("Number of links:%ld\n",nfile->st_nlink);
}
```

OUTPUT

```
Enter Filename
first.c
User Id: 1003
Blocksize: 1003
Last access time:1710342094
Last modification:1710342088
Production mode:33204
Size of file:137
Number of links:1
```

Program to implement opendir(). readdir(), closedir()

```
#include<stdio.h>
#include<dirent.h>
struct dirent *dptr;
int main(int argc,char *argv[])
{
char buff[256];
DIR *dirp;
printf("\n\nEnter directory name");
scanf("%s",buff);
if((dirp=opendir(buff))==NULL)
{
printf("Error");
exit(1);
}
while(dptr=readdir(dirp))
{
printf("%s\n",dptr->d_name);
}
closedir(dirp));
}
```

OUTPUT