Area of a circle (1)

```
echo "Enter the radious of the circle"
read r
area=$(echo "3.14 * $r* $r" | bc )
circum=$(echo "3.14*2*$r" | bc)
echo "area of the circle is   " $area
echo "circumference of the circle is " $circum
```

Output

```
Enter the radious of the circle
5
area of the circle is  78.50
circumference of the circle is 31.40
```

Calculator (2)

```
clear
sum=0
i="y"
echo "enter one no."
read n1
echo "enter second no."
read n2
while [ $i = "y" ]
do
echo "1.addition"
echo "2.substraction"
echo "3.multiplication"
echo "4.division"
echo "enter your choice"
read ch
case $ch in
1)sum=`expr $n1 + $n2`
echo "sum="$sum;;
2)sum=`expr $n1 - $n2`
echo "sub = "$sum;;
3)sum=`exp $n1 \* $n2`
echo "mul = "$sum;;
4)sum=`expr $n1 / $n2`
echo "div = "$sum;;
*)echo "invalid choice";;
esac
echo "do u want to continue ?"
read i
if [$i= "y" ]
then
xit
fi
done
```

Output

```
enter one no.
1
enter second no.
2
1.addition
2.substraction
3.multiplication
4.division
enter your choice
1
sum=3
do u want to continue ?
```

Greatest of 3 numbers(3)

```
echo "first num"
read a
echo "second num"
read b
echo "third num"
read c
m=$(( ($a + $b + $c) / 3))
p=`expr $a - $m`
q=`expr $b - $m`
r=`expr $c - $m`
d=$(( ($p * $p + $q * $q + $r * $r) / 3))
w=$(echo "sqrt ($d )" | bc)
echo "mean is $m"
echo "std dev $w"
```

Output

```
first num
1212
second num
122
third num
1321
mean is 885
std dev 541
```

Standard deviation(4)

```
echo "enter first number"
read a
echo "enter second number"
read b
echo " enter third number"
read c
m=$(( ( $a + $b + $c) / 3))
p=`expr $a - $m`
q=`expr $b - $m`
r=`expr $c - $m`
d=$(( ( $p * $p + $q * $q + $r * $r) / 3 ))
w=$(echo "sqrt ( $d )" | bc)
echo "mean of $a, $b, $c is $m"
echo "std dev is $w"
```

Output

```
enter first number
234
enter second number
1223
enter third number
12
mean of 234, 1223, 12 is 489
std dev is 526
```

Reverse of a number(5)

```
echo enter a number
read n
s=0
p=$n
while [ $n -gt 0 ]
do
r=$(($n % 10))
s=$(($s * 10 + $r))
n=$(($n / 10))
done
echo "rev of num $p is $s "
```

Output

```
enter a number
123
rev of num 123 is 321
```

Fibonacci series(6)

```
N-6
a=0
b=1
echo “the fibonacci series is :”
for ((i=0; i=n; i++))
do
echo -n “$a”
fn-$((a+b))
a-$b
b-$fn
done
```

Output

```
Fibonacci series is :
0
1
1
2
3
5
8
```

Armstrong(7)

```
echo enter a number
read n
s=0
p=$n
while [ $n -gt 0 ]
do
r=$(($n % 10))
s=$(($s + $r * $r * $r))
n=$(($n / 10))
done
if [ $p -eq $s ]
then
echo given number is amstrong
else
echo given number is not amstrong
fi
```

Output

```
enter a number
153
given number is amstrong
```

Factorial(8)

```
echo "enter a number"
read n
i=1
fact=1
while [ $i -le $n ]
do
fact=`expr $fact \* $i`
i=`expr $i + 1`
done
echo "factorial of number is $fact"
```

Output

```
enter a number
5
factorial of number is 120
```

Prime numbers upto the limit(9)

```
echo enter a limit
read limit
n=2
while [ $n -le $limit ]
do
i=2
f=0
while [ $i -lt $n ]
do
if [ `expr $n % $i` -eq 0 ]
then
f=1
break;
fi
i=`expr $i + 1`
done
if [ $f -eq 0 ]
then
echo $n
fi
n=`expr $n + 1`
done
```

output

```
enter a limit
10
2
3
5
7
```

Palindrom(10)

```
clear
echo enter a string
read a
b=`expr $a | wc -c`
b=`expr $b - 1`
echo number of letter=$b
while test $b -gt 0
do
e=`expr $a | cut -c $b`
d=$d$e
b=`expr $b - 1`
done
echo the reversed string is :$d
if test $a = $d
then
echo it is a palindrome
else
echo it is not a palindrome
fi
```

Output

```
enter a string
121
number of letter=3
the reversed string is :121
it is a palindrome
```

Counting words and lines(11)

```
echo enter the filename
read file
w=`cat $file | wc -w`
c=`cat $file | wc -c`
l=`grep -c "." $file`
echo number of charecters in $file is $c
echo number of word in $file is $w
echo number of lines in $file is $l
```

Output

```
enter the filename
area
number of charecters in area is 191
number of word in area is 31
number of lines in area is 6
```

Basic salary(12)

```
echo "enter ur basic salary "
read sal
da=`expr $sal \* 40 / 200`
ha=`expr $sal \* 20 / 100`
nsal=`expr $sal + $da + $ha`
echo "ur basic salary $sal "
echo "ur dearence allownce $da "
echo "ur house rent $ha "
echo "-------------- "
echo "ur net salary is Rs. $nsal "
```

Output

```
enter ur basic salary
100
ur basic salary 100
ur dearence allownce 20
ur house rent 20
--------------
ur net salary is Rs. 140
```

Command line argument(13)

```
(input the values during compilation of file)
sum=0
for i in $*
do
sum=`expr $sum + $i`
done
echo "summation of "$#" no. is: "$sum
avg=`expr $sum / $#`
echo "avg of "$#" no. is: "$avg
```

Output

```
ccst25@ccst25-OptiPlex-3010:~/Desktop$ sh comm 5 6 7
summation of 3 no. is: 18
avg of 3 no. Is: 6
```

Good morning and Good evening(14)

```
clear
hours=`date|cut -c12-13`
if [ $hours -le 12 ]
then
echo "good morning"
else
if [ $hours -le 16 ]
then
echo "good afternoon"
elif [ $hours -le 20 ]
then
echo "good evening"
else
echo "good night"
fi
fi
```

Output

```
Good afternoon
```

Test operator(15)

```
echo "enter a file name"
read file
if [ -e $file ]
then
echo "file exists"
if [ -r $file ]
then
echo "file has read access"
else
echo "file does not have read access"
fi
if [ -w $file ]
then
echo "file has write access"
else
echo "file does not have write access"
fi
if [ -x $file ]
then
echo "file has execute access"
else
echo "file does not have execute access"
fi
if [ -f $file ]
then
echo "file is an ordinary file"
else
echo "file is special file"
fi
if [ -d $file ]
then
echo "file is a directory"
else
echo "file is not a directory"
fi
if [ -s $file ]
then
echo "file size zero"
else
echo "file size not zero"
fi
fi
```

Output

```
enter a file name
pal
file exists
file has read access
file has write access
file does not have execute access
file is an ordinary file
file is not a directory
file size zero
```

<u>Vowels and consonents</u>(16)

```
echo "type any string"
read string
length=`echo $string | wc -c`
nvowels=0
nconsonants=0
ndigits=0
while [ $length -gt 1 ]
do
length=`expr $length - 1`
h=`echo $string | cut -c$length`
case $h in
[AaEeIiOoUu]) nvowels=`expr $nvowels + 1`
;;
[BbCcDdFfGgHhJjKkLlMmNnPpQqRrSsTtVvWwXxYyZz])
nconsonants=`expr $nconsonants + 1`
;;
[0-9]) ndigits=`expr $ndigits + 1`
;;
esac
done
echo "number of vowels:$nvowels"
echo "number of consonats:$nconsonants"
echo "number of digits:$ndigits"
```

Output

```
type any string

anjana

number of vowels:3

number of consonats:3

number of digits:0
```

Phonebook(17)

```
h=0
while [ $h -lt 5 ]
do clear
echo "1. add new record"
echo "2. display"
echo "3. search"
echo "4.delete"
echo "5. exit"
echo "enter yor choice"
read ch
case $ch in
1) echo enter name
read name
echo enter phone number
read phone
echo "$name $phone">>phonebook ;;
2) echo "name  phone"
cat phonebook;;
3) echo enter a name to search
read $name
nm=`grep $name phonebook | cut -f2`
echo "phone number of $sname is $nm";;
4) echo "enter a name to delete"
read nm
while readline
do
if [ `echo $line | cut -f1 -d' '` !=$nm ]
then
echo $line >> temp
fi
done < phonebook
cp temp phonebook
rm temp
echo record of $nm deleted;;
5) 'exit';;
esac
echo "do you want to continue"
read l
done
```

Output

```
1. add new record

2. display

3. search

4.delete

5. exit

enter yor choice

1

enter name

Yankee

enter phone number

9995559990

do you want to continue

2
```

Grade(18)

```
clear
echo -----------------------------------
echo '\tStudent Mark List'
echo -----------------------------------
echo Enter the Student name
read name
echo Enter the Register number
read rno
echo Enter the Mark1
read m1
echo Enter the Mark2
read m2
echo Enter the Mark3
read m3
echo Enter the Mark4
read m4
echo Enter the Mark5
read m5
tot=$(expr $m1 + $m2 + $m3 + $m4 + $m5)
avg=$(expr $tot / 5)
echo -----------------------------------
echo '\tStudent Mark List'
echo -----------------------------------
echo "Student Name  : $name"
echo "Register Number : $rno"
echo "Mark1: $m1"
echo "Mark2: $m2"
echo "Mark3: $m3"
echo "Mark4: $m4"
echo "Mark5: $m5"
echo "Total: $tot"
echo "Average: $avg"
if [ $avg -ge 90 ]
then
 echo "Grade: A+"
elif [ $avg -ge 80 ]
then
echo "Grade: A"
elif [ $avg -ge 70 ]
```

```
then
echo "Grade: B"
elif [ $avg -ge 60 ]
then
echo "Grade: C"
elif [ $avg -ge 50 ]
then
echo "Grade: D"
elif [ $avg -ge 35 ]
then
echo "Grade: E"
fi
```

Output

```
 Student Mark List
-----------------------------------
Enter the Student name
aaa
Enter the Register number
1
Enter the Mark1
50
Enter the Mark2
60
Enter the Mark3
60
Enter the Mark4
40
Enter the Mark5
30
-----------------------------------
 Student Mark List
-----------------------------------
Student Name : aaa
Register Number : 1
Mark1: 50
Mark2: 60
Mark3: 60
Mark4: 40
Mark5: 30
Total: 240
Average: 48
Grade : E
```

Sorting using arrays(19)

```
for i in {1..10}
do
echo “Enter a number :”
read a[$1]
done
for i in {1..19}
do
for j in (1..10)
do
if [${a[$1]} -it ${a[$j]}]]
then
t=${a[$i]}
a[$i]=${a[$]]}
a[$j]=$t
fi
done
done
echo ”sorted number”
echo ”=============”
fori in {1..10}
do
echo ${a[$i]}
done
```

Output

Date and time(20)

```
u=`who am i | cut -f1 -d' '`
d=`who am i |cut -f12 -d' '`
t=`who am i | cut -f13 -d' '`
echo "today is `date + %D`"
echo "current time is `date | cut -f5 -d' ' `"
echo "as of now `who | wc -l` user are login to system "
echo "my details............."
echo "user name $u"
echo "login date is $d"
echo "current login tym is $t"
```

Output

```
date: extra operand `%D'
Try `date --help' for more information.
today is
current time is 14:37:42
as of now 2 user are login to system
my details..............
user name ccst29
login date is 2019-02-04
current login tym is 13:50
```