**Shell Programming**

**1. Write a Shell program to check the given number is even or odd. PROGRAM**

echo “Enter a number:” read n

if [ `expr $n % 2` = 0 ] then

echo “Even number”

else

echo “Odd number”

fi

-bash-3.2$ sh evenodd.sh Enter a number:

6

Even number

**OUTPUT**

-bash-3.2$ sh evenodd.sh Enter a number:

83

Odd number

**2. Write a Shell program to check and display 10 leap years. PROGRAM**

for((i = 2000 ; i<= 2036 ; i++)) do

if [ `expr $i % 400` = 0 ] then

echo “$i is a leap year”

elif [ `expr $i % 4` = 0 -a `expr $i % 100` != 0 ] then

echo “$i is a leap year”

fi

done

**/ SL / CSE /     – 2**

**OUTPUT**

-bash-3.2$ sh leap.sh 2000 is a leap year 2004 is a leap year 2008 is a leap year 2012 is a leap year 2016 is a leap year 2020 is a leap year 2024 is a leap year 2028 is a leap year 2032 is a leap year 2036 is a leap year

**3. Write a Shell program to find the area and circumference of a circle.**

**PROGRAM**

echo “Enter the radius:” read r

area=`echo 3.14 \\* $r \\* $r | bc` cir=`echo 2 \\* 3.14 \\* $r | bc` echo “Area : $area”

echo “Circumference : $cir”

**OUTPUT**

-bash-3.2$ sh circle.sh Enter the radius:

3

Area : 28.26 Circumference : 18.84

**4. Write a Shell program to check the given number and its reverse are same.**

**PROGRAM**

echo “Enter a number:” read n

t=$n s=0

while [ $n -gt 0 ] do

r=`expr $n % 10` s=`expr $r + $s \\* 10` n=`expr $n / 10`

done

if [ $s = $t ] then

echo “The given number and its reverse are same”

else

echo “The given number and its reverse are not same”

fi

**OUTPUT**

-bash-3.2$ sh reverse.sh Enter a number:

123

The given number and its reverse are not same

-bash-3.2$ sh reverse.sh Enter a number:

121

The given number and its reverse are same

**5. Write a Shell program to check the given string is palindrome or not.**

**PROGRAM**

echo “Enter the string:” read s

l=`expr length $s` c=1

p=””

while [ $c -le $l ] do

e=`expr substr $s $c 1` p=$e$p

c=`expr $c + 1`

done

if [ $p = $s ] then

echo “The given string $s is a palindrome”

else

echo “The given string $s is not a palindrome”

fi

**OUTPUT**

-bash-3.2$ sh palindrome.sh Enter the string:

madam

The given string madam is a palindrome

-bash-3.2$ sh palindrome.sh Enter the string:

sir

The given string sir is not a palindrome

**6. Write a Shell program to find the sum of odd and even numbers from a set of numbers.**

**PROGRAM**

echo “Enter the number of elements:” read n

os=0

es=0

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

echo “Enter the number:” read no

if [ `expr $no % 2` = 0 ] then

es=`expr $es + $no`

else

os=`expr $os + $no`

fi

done

echo “The sum of odd numbers is : $os” echo “The sum of even numbers is : $es”

**OUTPUT**

-bash-3.2$ sh oddeven.sh Enter the number of elements: 5

Enter the number: 11

Enter the number: 22

Enter the number: 33

Enter the number: 44

Enter the number: 55

The sum of odd numbers is : 99 The sum of even numbers is : 66

**/ SL / CSE /     – 5**

**7. Write a Shell program to find the roots of a quadratic equation. PROGRAM**

echo “Enter the value for a” read a

echo “Enter the value for b” read b

echo “Enter the value for c” read c

d=`expr $b \\* $b – 4 \\* $a \\* $c`

x1=`echo “scale=3; (-$b + sqrt($d)) / (2 \* $a)” | bc` x2=`echo “scale=3; (-$b – sqrt($d)) / (2 \* $a)” | bc` echo “Root 1 : $x1”

echo “Root 2 : $x2”

**OUTPUT**

-bash-3.2$ sh quadratic.sh Enter the value for a

2

Enter the value for b 3

Enter the value for c 1

Root 1 : -.500

Root 2 : -1.000

**8. Write a Shell program to check the given integer is Armstrong number or not. PROGRAM**

echo “Enter a number:” read n

t=$n s=0

while [ $n -gt 0 ] do

r=`expr $n % 10`

s=`expr $s + $r \\* $r \\* $r` n=`expr $n / 10`

done

if [ $s = $t ] then

echo “$t is an armstrong number”

else

echo “$t is not an armstrong number”

fi

**/ SL / CSE /     – 6**

**OUTPUT**

-bash-3.2$ sh armstrong.sh Enter a number:

123

123 is not an armstrong number

-bash-3.2$ sh armstrong.sh Enter a number:

153

153 is an armstrong number

**9. Write a Shell program to check the given integer is prime or not.**

**PROGRAM**

echo “Enter a number:” read n

flag=0

for((i = 2 ; i <= n / 2 ; i++)) do

r=`expr $n % $i` if [ $r = 0 ] then

flag=1 break

fi

done

if [ $flag = 0 ] then

echo “$n is a prime number”

else

echo “$n is not a prime number”

fi

**OUTPUT**

-bash-3.2$ sh prime.sh Enter a number:

5

5 is a prime number

-bash-3.2$ sh prime.sh Enter a number:

10

10 is not a prime number

**/ SL / CSE /     – 7**

**10. Write a Shell program to generate prime numbers between 1 and 50.**

**PROGRAM**

for((n = 1 ; n <= 50 ; n++)) do

flag=0

for((i = 2 ; i <= n / 2; i++)) do

r=`expr $n % $i` if [ $r = 0 ] then

flag=1 break

fi

done

if [ $flag = 0 ] then

echo $n

fi

done

**OUTPUT**

-bash-3.2$ sh genprime.sh 1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

**/ SL / CSE /     – 8**

**11. Write a Shell program to find the sum of square of individual digits of a number.**

**PROGRAM**

echo “Enter a number:” read n

t=$n s=0

while [ $n -gt 0 ] do

r=`expr $n % 10` s=`expr $s + $r \\* $r` n=`expr $n / 10`

done

echo “The sum of square of individual digits of $t is $s”

**OUTPUT**

-bash-3.2$ sh square.sh Enter a number:

124

The sum of square of individual digits of 124 is 21

**12. Write a Shell program to find the sum of cube of individual digits of a number.**

**PROGRAM**

echo “Enter a number:” read n

t=$n s=0

while [ $n -gt 0 ] do

|  |  |  |
| --- | --- | --- |
| r=`expr $n | | % 10` |
| s=`expr | $s | + $r \\* $r \\* $r` |
| n=`expr | $n | / 10` |

done

echo “The sum of cube of individual digits of $t is $s”

**OUTPUT**

-bash-3.2$ sh cube.sh Enter a number:

124

The sum of cube of individual digits of 124 is 73

**/ SL / CSE /     – 9**

**13. Write a Shell program to execute various UNIX commands using case statements set of numbers.**

**PROGRAM**

echo “1-who am I?”

echo “2-who is logged on?” echo “3-date”

echo “4-calendar”

echo “Enter your choice:” read n

case $n in

1)whoami ;;

2)who ;;

3)date ;;

4)cal ;;

esac

**OUTPUT**

-bash-3.2$ sh commands.sh 1-who am I?

2-who is logged on?

3-date

4-calendar

Enter your choice: 1

bhuvan

**14. Write a Shell program to count the number of vowels in a line of text.**

**PROGRAM**

echo “Enter the text:” read s

l=`expr length $s` c=1

vc=0

while [ $c -le $l ] do

r=`expr substr $s $c 1`

if [ $r = ‘a’ -o $r = ‘e’ -o $r = ‘i’ -o $r = ‘o’ -o $r = ‘u’ ] then

vc=`expr $vc + 1`

fi

c=`expr $c + 1` done

echo “The number of vowels in the text $s is : $vc”

**/ SL / CSE /     – 10**

**OUTPUT**

-bash-3.2$ sh vowels.sh Enter the text: computer

The number of vowels in the text computer is : 3

**15. Write a Shell program to display student grades.**

**PROGRAM**

echo “Enter the number of students:” read n

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

echo “Enter roll no.:” read rollno

echo “Enter name:” read name

echo “Entermark-1” read m1

echo “Entermark-2:” read m2

echo “Entermark-3:” read m3

tot=`expr $m1 + $m2 + $m3` avg=`expr $tot / 3`

if [ $avg -ge 75 ] then

grade=”Distinction” elif [ $avg -ge 60 ]

then

grade=”First class” elif [ $avg -ge 50 ]

then

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| grade=”Second class” | | |  |  |
| else |  |  |  |  |
| grade=”Fail” | |  |  |  |
| fi | Name | Total | Average | Grade” |
| echo “Roll no. |
| echo “$rollno | $name | $tot | $avg | $grade” |

done

**/ SL / CSE /     – 11**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OUTPUT** |  |  |  |  |
| -bash-3.2$ sh grades.sh | |  |  |  |
| Enter the number of students: | | |  |  |
| 2 |  |  |  |  |
| Enter roll no.: |  |  |  |  |
| 201101001 |  |  |  |  |
| Enter name: |  |  |  |  |
| Arun |  |  |  |  |
| Enter mark-1 |  |  |  |  |
| 75 |  |  |  |  |
| Enter mark-2: |  |  |  |  |
| 60 |  |  |  |  |
| Enter mark-3: |  |  |  |  |
| 65 | Name | Total | Average | Grade |
| Roll no. |
| 201101001 | Arun | 200 | 66 | First class |
| Enter roll no.: |  |  |  |  |
| 201101002 |  |  |  |  |
| Enter name: |  |  |  |  |
| Bhuvan |  |  |  |  |
| Enter mark-1 |  |  |  |  |
| 85 |  |  |  |  |
| Enter mark-2: |  |  |  |  |
| 90 |  |  |  |  |
| Enter mark-3: |  |  |  |  |
| 85 | Name | Total | Average | Grade |
| Roll no. |
| 201101002 | Bhuvan | 260 | 86 | Distinction |

**16. Write a Shell program to find the smallest number from a set of numbers.**

**PROGRAM**

echo “Enter the number of elements:” read n

s=999999

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

echo “Enter the number:” read no

if [ $no -lt $s ] then

s=$no

fi

done

echo “The smallest number is : $s”

**/ SL / CSE /     – 12**

**OUTPUT**

-bash-3.2$ sh smallest.sh Enter the number of elements: 5

Enter the number: 22

Enter the number: 33

Enter the number: 11

Enter the number: 44

Enter the number: 55

The smallest number is : 11

**17. Write a Shell program to find the smallest digit from a number.**

**PROGRAM**

echo “Enter a number:” read n

s=9

while [ $n -gt 0 ] do

r=`expr $n % 10` if [ $r -lt$s ] then

s=$r

fi

n=`expr $n / 10`

done

echo “The smallest digit is : $s”

**OUTPUT**

-bash-3.2$ sh small.sh Enter a number:

143

The smallest digit is : 1

-bash-3.2$ sh small.sh Enter a number:

786

The smallest digit is : 6

**/ SL / CSE /     – 13**

**18. Write a Shell program to find the sum of all numbers between 50 and 100, which are divisible by 3 and not divisible by 5.**

**PROGRAM**

for((i = 50 ; i<= 100 ; i++)) do

if [ `expr $i % 3` = 0 -a `expr $i % 5` != 0 ] then

echo $i

fi

done

**OUTPUT**

-bash-3.2$ sh divisible.sh 51 54 57 63 66 69 72 78 81 84 87 93 96 99

**19. Write a Shell program to find the sum of digits of a number until a single digit is obtained.**

**PROGRAM**

echo “Enter a number:” read n

s=0

while [ $n -gt 0 ] do

r=`expr $n % 10` s=`expr $s + $r` n=`expr $n / 10`

if [ $n = 0 -a $s -gt 9 ] then

n=$s s=0

fi

done

echo “The single digit sum is : $s”

**/ SL / CSE /     – 14**

**OUTPUT**

-bash-3.2$ sh digitsum.sh Enter a number:

14

The single digit sum is : 5

-bash-3.2$ sh digitsum.sh Enter a number:

1983

The single digit sum is : 3

**20. Write a Shell program to find the second highest number from a set of numbers.**

**PROGRAM**

echo “Enter the number of elements:” read n

a=0

b=0

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

echo “Enter the number:” read no

if [ $no -gt $a ] then

b=$a a=$no

elif [ $no -gt $b ] then

b=$no

fi

done

echo “The second highest number is : $b”

**OUTPUT**

Enter the number of elements: 5

Enter the number: 11

Enter the number: 22

Enter the number: 33

Enter the number: 44

Enter the number: 55

The second highest number is : 44

**/ SL / CSE /     – 15**

**21. Write a Shell program to find the second largest digit from a number.**

**PROGRAM**

echo “Enter a number:” read n

a=0

b=0

while [ $n -gt 0 ] do

r=`expr $n % 10` if [ $r -gt$a ] then

b=$a a=$r

elif [ $r -gt $b ] then

b=$r

fi

n=`expr $n / 10`

done

echo “The second largest digit is : $b”

**OUTPUT**

-bash-3.2$ sh seclarge.sh Enter a number:

1983

The second largest digit is : 8

**22. Write a Shell program to find the sum of odd digits and even digits from a number.**

**PROGRAM**

echo “Enter a number:” read n

os=0

es=0

while [ $n -gt 0 ] do

r=`expr $n % 10`

if [ `expr $r % 2` = 0 ] then

es=`expr $es + $r`

else

os=`expr $os + $r`

fi

n=`expr $n / 10`

done

**/ SL / CSE /     – 16**

echo “The sum of odd digits is : $os” echo “The sum of even digits is : $es”

**OUTPUT**

Enter a number: 1988

The sum of odd digits is : 10 The sum of even digits is : 16

**23. Write a Shell program to find the sum of two numbers using function programming.**

**PROGRAM**

-bash-3.2$ sum()

>{

>echo `expr $1 + $2`

>}

**OUTPUT**

-bash-3.2$ sum 10 20 30

**24. Write a Shell program to find the largest number between two numbers using function.**

**PROGRAM**

-bash-3.2$ largest()

>{

>if [ $1 -gt $2 ]

>then

>echo “$1 is greater”

>else

>echo “$2 is greater”

>fi

>}

**OUTPUT**

-bash-3.2$ largest 10 20 20 is greater

-bash-3.2$ largest 20 10 20 is greater

**/ SL / CSE /     – 17**

**25. Write a Shell program to find the largest among three numbers.**

**PROGRAM**

echo “Enter the first number:” read a

echo “Enter the second number:” read b

echo “Enter the third number:” read c

if [ $a -gt $b -a $a -gt $c ] then

echo “$s is greater” elif [ $b -gt $c ]

then

echo “$b is greater”

else

echo “$c is greater”

fi

**OUTPUT**

-bash-3.2$ sh larthree.sh Enter the first number: 20

Enter the second number: 30

Enter the third number: 10

30 is greater-bash-3.2$

**26. Write a Shell program to find the largest among ‘n’ different numbers.**

**PROGRAM**

echo “Enter the number of elements:” read n

l=0

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

echo “Enter the number:” read no

if [ $no -gt $l ] then

l=$no

fi

done

echo “The largest numbers is : $l”

**OUTPUT**

-bash-3.2$ sh largest.sh Enter the number of elements: 5

Enter the number: 44

Enter the number: 55

Enter the number: 33

Enter the number: 22

Enter the number: 11

The largest numbers is : 55

**27. Write a Shell program to find the largest digit of a number.**

**PROGRAM**

echo “Enter a number:” read n

s=0

while [ $n -gt 0 ] do

r=`expr $n % 10` if [ $r -gt$s ] then

s=$r

fi

n=`expr $n / 10`

done

echo “The largest digit is : $s”

**OUTPUT**

-bash-3.2$ sh large.sh Enter a number:

143

The largest digit is : 4

-bash-3.2$ sh large.sh Enter a number:

786

The largest digit is : 8

**/ SL / CSE /     – 19**

**28. Write a Shell program to find the sum of ‘n’ different numbers.**

**PROGRAM**

echo “Enter the number of elements:” read n

s=0

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

echo “Enter the number:” read no

s=`expr $s + $no`

done

echo “The sum is : $s”

**OUTPUT**

-bash-3.2$ sh sum.sh

Enter the number of elements: 5

Enter the number: 11

Enter the number: 22

Enter the number: 33

Enter the number: 44

Enter the number: 55

The sum is : 165

**29. Write a Shell program to find the sum of digits of a number.**

**PROGRAM**

echo “Enter a number:” read n

s=0

while [ $n -gt 0 ] do

r=`expr $n % 10` s=`expr $s + $r` n=`expr $n / 10`

done

echo “The sum of digit is : $s”

**/ SL / CSE /     – 20**

**OUTPUT**

-bash-3.2$ sh sumdigit.sh Enter a number:

14

The sum of digit is : 5-bash-3.2$ sh sumdigit.sh Enter a number:

1983

The sum of digit is : 21

**30. Write a Shell program to print the reverse of a number.**

**PROGRAM**

echo “Enter a number:” read n

t=$n s=0

while [ $n -gt 0 ] do

r=`expr $n % 10` s=`expr $r + $s \\* 10` n=`expr $n / 10`

done

echo “The reverse of the number $t is $s”

**OUTPUT**

-bash-3.2$ sh revnum.sh Enter a number:

123

The reverse of the number 123 is 321

**31. Write a Shell program to find the factorial of a number using for loop.**

**PROGRAM**

echo “Enter a number:” read n

f=1

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

f=`expr $f \\* $i`

done

echo “The factorial of $n is $f”

**OUTPUT**

-bash-3.2$ sh factorial.sh Enter a number:

5

The factorial of 5 is 120

**32. Write a Shell program to generate Fibonacci series.**

**PROGRAM**

echo “Enter the number of terms:” read n

echo “Fibonacci series is:” a=-1

b=1

c=0

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

c=`expr $a + $b` echo $c

a=$b b=$c

done

**OUTPUT**

-bash-3.2$ sh fibonacci.sh Enter the number of terms: 5

Fibonacci series is: 0 1 1 2 3