ITOM ASSIGNMENT -1

1. s=”india1234567890”

echo $s

echo “${s//[!0-9]/}”

echo “${s//[[:digit:]]/}”

echo “${s^^}”

sample ip/op:

india1234567890

1234567890

india

INDIA1234567890

2. cat >a.txt

Apple

Orange

banana

cat >b.txt

cp a.txt b.txt

cat b.txt

sample ip/op:

b.txt

Apple

Orange

Banana

3. ls -l /etc/passwd

Sample ip/op:

-rw-r--r-- 1 root root 1501 May 11 16:58 /etc/passwd

4. cat /etc/shadow

5. cat/etc/passwd

6.useradd -u 1100 ami //with specific user id

Useradd -u 1001 -g 1001 ami //with specific group id

Useradd -m ami //add user without home directory

7. Useradd -G groupname username //add user to group and again the same to

add another user

sudo gpasswd -d username groupname //to remove user in a group

sudo groupdel groupname //to del group

deluser username

8.

#!/bin/sh

# --------------------

# Stack implementation

# --------------------

s=""

stk=""

STACK\_MAX\_SIZE="65536"

# Delete all values from the stack:

stack\_clear () {

s=""

stk=""

}

# To push a value into the stack:

stack\_push () {

local counter

local cnt

counter=$(echo -n "${s}" | wc --bytes)

cnt=$(echo -n "${counter}" | wc --bytes)

# ----- Internal check begin -----

check=$(echo -n "${cnt}" | wc --bytes)

if test "${check}" != "1"

then

echo "Internal error: it is real to meet such a long string..."

exit 2s

fi

# ----- Internal check end -----

stk=$(echo -n "${stk}${s}${counter}${cnt}")

local check

check=$(echo -n "${stk}" | wc --bytes)

if test "${check}" -gt "${STACK\_MAX\_SIZE}"

then

echo "Error: stack overflow."

exit 1

fi

}

# To pull a value from the stack:

stack\_pop () {

local counter

local cnt

if test "${stk}" = ""

then

echo "Error: trying to pop from an empty stack."

exit 1

fi

cnt=$(echo -n "${stk}" | tail --bytes=1)

stk=$(echo -n "${stk}" | head --bytes=-1)

counter=$(echo -n "${stk}" | tail --bytes=${cnt})

stk=$(echo -n "${stk}" | head --bytes=-${cnt})

s=$(echo -n "${stk}" | tail --bytes=${counter})

stk=$(echo -n "${stk}" | head --bytes=-${counter})

# ----- Internal check begin -----

local check

check=$(echo -n "${s}" | wc --bytes)

if test "${check}" != "${counter}"

then

echo "Internal error: the stack is damaged."

exit 2

fi

# ----- Internal check end -----

}

# ---------------

# The entry point

# ---------------

# Push "one", "two", "three" into the stack:

s="one"; stack\_push

s="two"; stack\_push

s="three"; stack\_push

# Extract all the data from the stack:

while test "${stk}" != ""

do

stack\_pop

echo "${s}"

done

I/O:

one

two

three

9.

Search element :

read search

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

do

If(I -eq search)

do

Echo search element is present in stack

else

Echo search element is not  present in stack

End if

End

O/P:

One

Element present in stack

SIZE OF STACK:

${#stack[\*]}

O/P:

3

10.var=”apple”

function fruit

{

Local var=”orange”

Echo $var

}

echo $var

fruit

echo $var

sample ip/op:

apple

apple

orange

apple

11.

# ls directory\_name | wc -l //count no. of files in a directory

12.

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.Subtraction"

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=`expr $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

exit

fi

done

sample ip/op:

$ sh calculator.sh

Enter any no.

121

Enter one no.

21

Enter second no.

58

1.Addition

2.Subtraction

3.Multiplication

4.Division

Enter your choice

1

Sum =79

13.

Var =”ab”

array=(ab bc cd ef)

for item in “${array[@]}”;do

[[ $var ==”$item” ]] && echo “$var present in array”

Done

14.

echo -n “enter number:”

read n

num=$(( n % 2 ))

if [ $num -eq 0 ]

then

echo “$n is even number”

else

echo “$n is odd number”

fi

sample ip/op:

5

5 is odd number

15.

var=”some string”

echo ${#var}

echo $var | rev

a=”abc”

b=”def”

if [ “$a” == “$b” ]; then

echo “match”

else

echo “don’t match”

fi

sample ip/op:

gnirts emos

don’t match