PRACTICAL 4

AIM: Write a shell script to validate the entered date. (e.g., Date format is: dd-mm-yyyy)

Algorithm:

- Enter date in DD/MM/YYYY Format.
- Check year validation, if year is not valid print error.
- If year is valid, check month validation (i.e., month is between 1 to 12), if month is not valid print error.
- If month is valid, then finally check day validation with leap year condition, here we will day range from 1 to 30, 1 to 31, 1 to 28 and 1 to 29.
- If day is valid print date is correct otherwise print error.

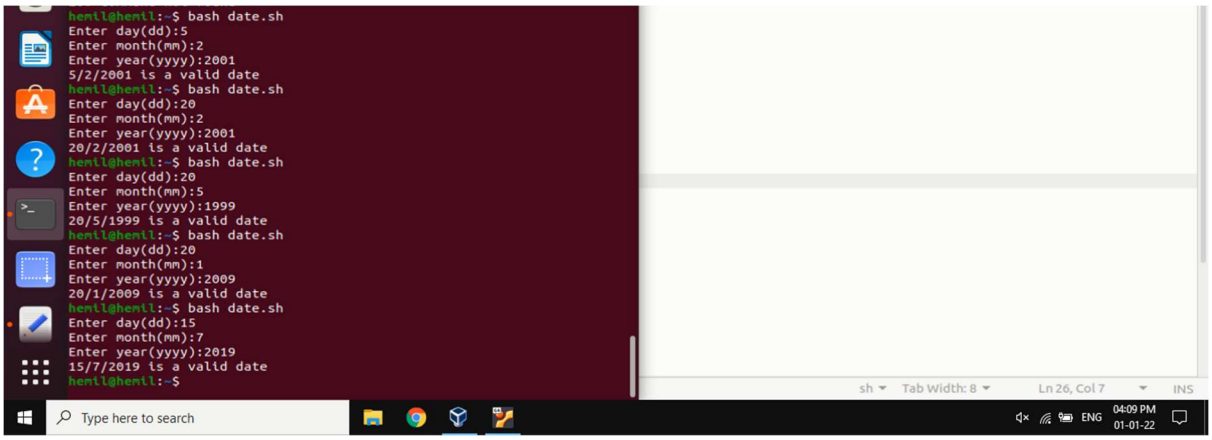
Code:

```
dd=0
mm=0
yy=0
days=0
echo -n "Enter day(dd):"
read dd
echo -n "Enter month(mm):"
read mm
echo -n "Enter year(yyyy):"
read yy
if [ $mm -le 0 -o $mm -gt 12 ];
then
echo "$mm is invalid month."
exit 1
fi
case $mm in
1) days=31;;
2) days=28;;
3) days=31;;
4) days=30;;
5) days=31;;
6) days=30;;
7) days=31;;
8) days=31;;
```

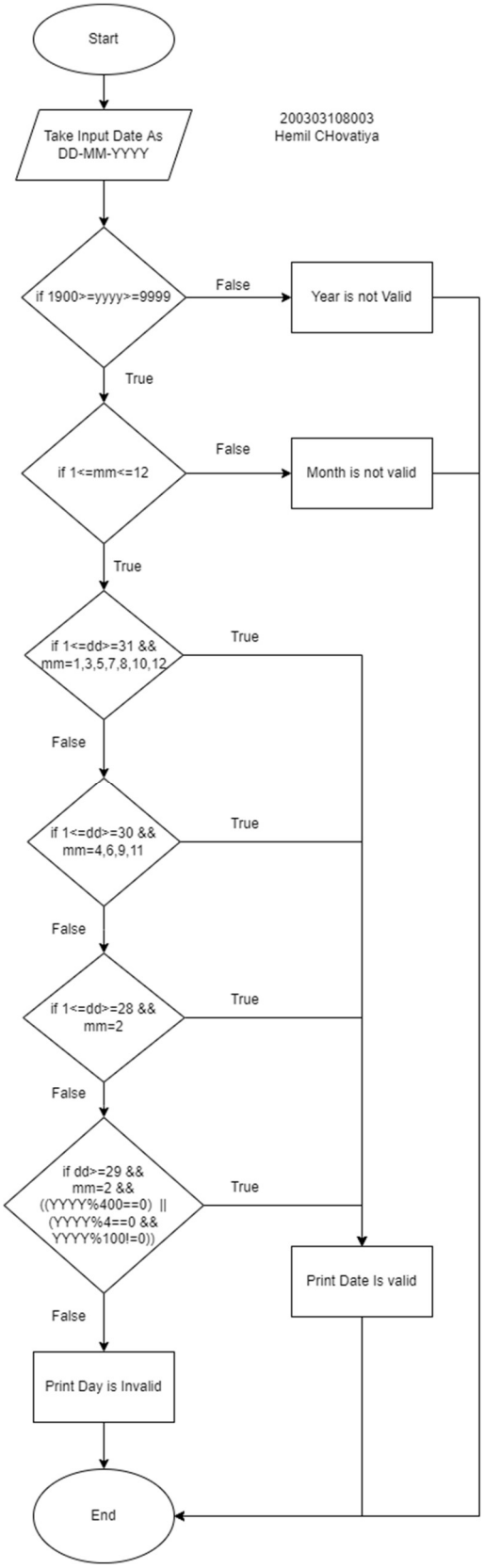
```
9) days=30;;
10) days=31;;
11) days=30;;
12) days=31;;
*) days=-1;;
esac
if [ $mm -eq 2 ];
then
if [ $((yy % 4)) -ne 0 ] ; then
:
elif [ $(yy % 400) -eq 0 ] ; then
days=29

elif [ $((yy % 100)) -eq 0 ] ; then
:
else
days=29
fi
fi
if [ $dd -le 0 -o $dd -gt $days ];
then
echo "$dd day is invalid"
exit 3
fi
echo "$dd/$mm/$yy is a valid date"
```

Output:



Flowchart:


Start
Take Input Date As DD-MM-YYYY
200303108003
Hemil CHovatiya
if 1900>=yyyy>=9999
False
Year is not Valid
True
if 1<=mm<=12
False
Month is not valid
True
if 1<=dd>=31 && mm=1,3,5,7,8,10,12
True
False
if 1<=dd>=30 && mm=4,6,9,11
True
False
if 1<=dd>=28 && mm=2
True
False
if dd>=29 && mm=2 && ((YYYY%400==0) || (YYYY%4==0 && YYYY%100!=0))
True
False
Print Date Is valid
Print Day is Invalid
End

PRACTICAL 5

AIM: Write a shell script to check entered string is palindrome or not

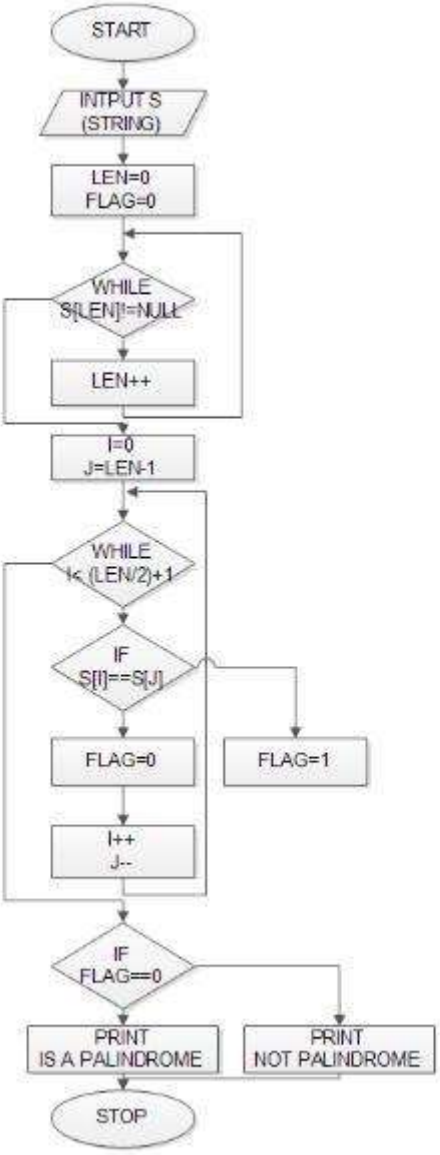
Algorithm:

- Input a String
- Initialize Len to zero , Flag to zero
- While String[Len] is not equal to NULL
- Increment Len
- Initialize I to zero , J to Len-1
- If va1 equal to rev
- Print Key Is a Palindrome
- else
- Print Key Is Not a Palindrome
- Stop

Detailed Algorithm:

```
Step 1:  Input S (string)
Step 2:  Len = 0 , Flag =0
Step 3:  While (S[Len] != NULL)
                Len++
Step 4:  I = 0 , J = Len-1
Step 5:  While ( I < (Len/2)+1 )
                If ( S[I] == S[J] )
                        Flag=0
                else
                        Flag=1
                I++ , J–
Step 6:     If ( Flag == 0 )
                        Print Key Is a Palindrome
              else
                        Print Key Is Not a Palindrome
Step 7: End
```

Flow Chart:


START
INTPUT S (STRING)
LEN=0 FLAG=0
WHILE S[LEN]!=NULL
LEN++
I=0 J=LEN-1
WHILE I< (LEN/2)+1
IF S[I]==S[J]
FLAG=0
FLAG=1
I++ J--
IF FLAG==0
PRINT IS A PALINDROME
PRINT NOT PALINDROME
STOP

Code:

```
echo Enter the value of string
read s
echo $s>temp
rvs="$(rev temp)"
if [ $s = $rvs ]
then
echo "it is palindrome"
else
echo " it is not a Palindrome"
fi
```

Output: