**USP Lab**

**Week 8**

1. Write a script where a password is stored in a variable. The program allows user to enter the password and checks for it. The script has to be executed until a correct guess is made.
2. Write a shell script that accepts filenames as arguments and prints them in the reverse order.
3. Write a shell script to check whether argument is positive or negative number and reports an error if a number is not given.
4. Write a shell script to check whether the given number is a palindrome or not.
5. Write a shell script that accepts a filename, starting and ending line numbers as arguments and displays all the lines between the given line numbers

|  |  |
| --- | --- |
| **Answers** | |
| **1.**  password="open"  answer=""  until test "$answer" = "$password"  do  echo "Guess the password to quit the program"  read answer  done  **2.**  for ((i=$#;i>0;i--)  do  echo $i  done  **3.**  if [ $1 -gt 0 ]; then  echo "$1 is positive"  elif [ $1 -lt 0 ]  then  echo "$1 is negative"  elif [ $1 -eq 0 ]  then  echo "$1 is zero"  else  echo "Opps! $1 is not number, give number"  fi | **4.**  echo "Enter the no:" read n sd=0 rev=" " on=$n while [ $n -gt 0 ] do sd=$(( $n % 10 )) n=$(( $n / 10 )) rev=$( echo ${rev}${sd} ) done =$(( $n / 10 )) rev=$( echo ${rev}${sd} ) else echo " Not palindrome" fi  **5**.  if test $# -ne 3  then  echo “usage: filename <starting no><ending no>”  else  lastline=`wc –l < $1`  if test $2 –lt $lastline –a $3 –le $lastline  then  nline=`expr $3 - $2 + 1`  echo “`tail +$2 $1|head -$nline`”  else  echo “invalid range specification”  fi  fi |