#!/bin/bash

# Introducing shift, &&, and |, stdout redirection 1>&
# is_odd()
{
    [ $# == 0 ] && echo "usage: call $0 with 1 argument" 1>& 2
    remainder=$(($1 % 2))
    if [ $remainder == 1 ]; then
        return 0
    else
        return 1
    fi
}

while [ $# −ne 0 ]; do
    if is_odd $1; then
        echo "$1 is odd"
    else
        echo "$1 is even"
    fi
    shift
done

# Introducing :, $RANDOM, echo -e
# guess=$RANDOM
# count=0
while :
    do
        read −p "guess: " −e num
        if [ $num −lt $guess ]; then
            echo −e "\low"
        elif [ $num −gt $guess ]; then
            echo −e "\high"
        else
            echo −e "\bingo. you have taken $count guess"
            break
        fi
        count=$(($count+1))
done
#!/bin/bash
#
# You can redirect input and output to function
# and while loops too.
#
mycat()
{
  while read name userid rest;
do
    if [ $userid != $2 ]; then
      echo $name $userid $rest
    fi
  done < $1
}

mycat $1 $2 > tmp
mv tmp $1

#!/bin/sh
#
# Introducing sed, the stream editor
#
for i in "$*"; do
  newname=$(echo $i | sed 's/ /_/g')
  mv "$i" $newname
done
#!/bin/sh
# More sed.
if [ $# -ne 1 ]; then
  echo "usage: $0 userid" 1>&2
  exit
fi
sed /$1/d < student.db

#!/bin/bash
#
# Introducing awk.
#
# "q2 find userid" in one line of awk and sed!
#
# Note (i) the different between $n meant for awk
# and $n meant for bash (ii) the meaning for $0
# is different for awk and bash
# cat student.db | awk "\$2 ~/\$1/ { print \$0 }" | sed 's/_/ /g'
#!/bin/bash
#
# More awk
#
awk 'BEGIN {
    w1 = 5.0/10
    w2 = 8.0/10
    w3 = 7.0/10
    print "AVERAGE FOR CS2281"
    }
    {
        total_a1 += $2
        total_a2 += $3
        total_a3 += $4
        count++
        printf "%s %.1f\n", $1, $2*w1+$3*w2+$4*w3
    }
    END {
        printf "A1 = %.1f, A2 = %1.f, A3 = %1.f\n", total_a1/count, total_a2/count, total_a3/count
        printf "AVERAGE = %.1f\n", (total_a1*w1/count)+(total_a2*w2/count)+(total_a3*w3/count)
    }' < assignment.db