SORTING PROGRAM

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$ nano sort.sh

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$ bash sort.sh

797 is 2nd largest value

408 is 2nd smallest value

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$ nano sort.sh

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$ bash sort.sh

832 is 2nd largest value

306 is 2nd smallest value

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$ cat sort.sh

for (( i=0; i<10; i++ ))

do

random\_number=$(($(($RANDOM%900))+100))

arr[$i]=$random\_number

done

max\_1=0

max\_2=0

for (( i=0; i<10; i++ ))

do

if [ ${arr[i]} -gt $max\_1 ]

then

max\_2=$max\_1

max\_1=${arr[i]}

elif [ ${arr[i]} -gt $max\_2 -a ${arr[i]} -ne $max\_1 ]

then

max\_2=${arr[i]}

fi

done

min\_1=$max\_2

for (( i=0; i<10; i++ ))

do

if [ ${arr[i]} -lt $min\_1 ]

then

min\_2=$min\_1

min\_1=${arr[i]}

elif [ ${arr[i]} -lt $min\_2 -a ${arr[i]} -ne $min\_1 ]

then

min\_2=${arr[i]}

fi

done

echo $max\_2 " is 2nd largest value"

echo $min\_2 " is 2nd smallest value"

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$ bash -x sort.sh

+ (( i=0 ))

+ (( i<10 ))

+ random\_number=610

+ arr[$i]=610

+ (( i++ ))

+ (( i<10 ))

+ random\_number=545

+ arr[$i]=545

+ (( i++ ))

+ (( i<10 ))

+ random\_number=111

+ arr[$i]=111

+ (( i++ ))

+ (( i<10 ))

+ random\_number=393

+ arr[$i]=393

+ (( i++ ))

+ (( i<10 ))

+ random\_number=858

+ arr[$i]=858

+ (( i++ ))

+ (( i<10 ))

+ random\_number=938

+ arr[$i]=938

+ (( i++ ))

+ (( i<10 ))

+ random\_number=432

+ arr[$i]=432

+ (( i++ ))

+ (( i<10 ))

+ random\_number=544

+ arr[$i]=544

+ (( i++ ))

+ (( i<10 ))

+ random\_number=966

+ arr[$i]=966

+ (( i++ ))

+ (( i<10 ))

+ random\_number=917

+ arr[$i]=917

+ (( i++ ))

+ (( i<10 ))

+ max\_1=0

+ max\_2=0

+ (( i=0 ))

+ (( i<10 ))

+ '[' 610 -gt 0 ']'

+ max\_2=0

+ max\_1=610

+ (( i++ ))

+ (( i<10 ))

+ '[' 545 -gt 610 ']'

+ '[' 545 -gt 0 -a 545 -ne 610 ']'

+ max\_2=545

+ (( i++ ))

+ (( i<10 ))

+ '[' 111 -gt 610 ']'

+ '[' 111 -gt 545 -a 111 -ne 610 ']'

+ (( i++ ))

+ (( i<10 ))

+ '[' 393 -gt 610 ']'

+ '[' 393 -gt 545 -a 393 -ne 610 ']'

+ (( i++ ))

+ (( i<10 ))

+ '[' 858 -gt 610 ']'

+ max\_2=610

+ max\_1=858

+ (( i++ ))

+ (( i<10 ))

+ '[' 938 -gt 858 ']'

+ max\_2=858

+ max\_1=938

+ (( i++ ))

+ (( i<10 ))

+ '[' 432 -gt 938 ']'

+ '[' 432 -gt 858 -a 432 -ne 938 ']'

+ (( i++ ))

+ (( i<10 ))

+ '[' 544 -gt 938 ']'

+ '[' 544 -gt 858 -a 544 -ne 938 ']'

+ (( i++ ))

+ (( i<10 ))

+ '[' 966 -gt 938 ']'

+ max\_2=938

+ max\_1=966

+ (( i++ ))

+ (( i<10 ))

+ '[' 917 -gt 966 ']'

+ '[' 917 -gt 938 -a 917 -ne 966 ']'

+ (( i++ ))

+ (( i<10 ))

+ min\_1=938

+ (( i=0 ))

+ (( i<10 ))

+ '[' 610 -lt 938 ']'

+ min\_2=938

+ min\_1=610

+ (( i++ ))

+ (( i<10 ))

+ '[' 545 -lt 610 ']'

+ min\_2=610

+ min\_1=545

+ (( i++ ))

+ (( i<10 ))

+ '[' 111 -lt 545 ']'

+ min\_2=545

+ min\_1=111

+ (( i++ ))

+ (( i<10 ))

+ '[' 393 -lt 111 ']'

+ '[' 393 -lt 545 -a 393 -ne 111 ']'

+ min\_2=393

+ (( i++ ))

+ (( i<10 ))

+ '[' 858 -lt 111 ']'

+ '[' 858 -lt 393 -a 858 -ne 111 ']'

+ (( i++ ))

+ (( i<10 ))

+ '[' 938 -lt 111 ']'

+ '[' 938 -lt 393 -a 938 -ne 111 ']'

+ (( i++ ))

+ (( i<10 ))

+ '[' 432 -lt 111 ']'

+ '[' 432 -lt 393 -a 432 -ne 111 ']'

+ (( i++ ))

+ (( i<10 ))

+ '[' 544 -lt 111 ']'

+ '[' 544 -lt 393 -a 544 -ne 111 ']'

+ (( i++ ))

+ (( i<10 ))

+ '[' 966 -lt 111 ']'

+ '[' 966 -lt 393 -a 966 -ne 111 ']'

+ (( i++ ))

+ (( i<10 ))

+ '[' 917 -lt 111 ']'

+ '[' 917 -lt 393 -a 917 -ne 111 ']'

+ (( i++ ))

+ (( i<10 ))

+ echo 938 ' is 2nd largest value'

938 is 2nd largest value

+ echo 393 ' is 2nd smallest value'

393 is 2nd smallest value

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$

TOSSING A COIN

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$ nano while.sh

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$ bash while.sh

1 head

1 tail

2 head

2 tail

3 head

3 tail

4 head

4 tail

5 head

5 tail

6 head

6 tail

7 head

7 tail

8 head

8 tail

9 head

9 tail

10 head

10 tail

11 head

11 tail

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$ nano while.sh

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$ cat while.sh

i=1

while [ $i -lt 12 ]

do

echo "$i head"

echo "$i tail"

((i++))

done

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$ bash -x while.sh

+ i=1

+ '[' 1 -lt 12 ']'

+ echo '1 head'

1 head

+ echo '1 tail'

1 tail

+ (( i++ ))

+ '[' 2 -lt 12 ']'

+ echo '2 head'

2 head

+ echo '2 tail'

2 tail

+ (( i++ ))

+ '[' 3 -lt 12 ']'

+ echo '3 head'

3 head

+ echo '3 tail'

3 tail

+ (( i++ ))

+ '[' 4 -lt 12 ']'

+ echo '4 head'

4 head

+ echo '4 tail'

4 tail

+ (( i++ ))

+ '[' 5 -lt 12 ']'

+ echo '5 head'

5 head

+ echo '5 tail'

5 tail

+ (( i++ ))

+ '[' 6 -lt 12 ']'

+ echo '6 head'

6 head

+ echo '6 tail'

6 tail

+ (( i++ ))

+ '[' 7 -lt 12 ']'

+ echo '7 head'

7 head

+ echo '7 tail'

7 tail

+ (( i++ ))

+ '[' 8 -lt 12 ']'

+ echo '8 head'

8 head

+ echo '8 tail'

8 tail

+ (( i++ ))

+ '[' 9 -lt 12 ']'

+ echo '9 head'

9 head

+ echo '9 tail'

9 tail

+ (( i++ ))

+ '[' 10 -lt 12 ']'

+ echo '10 head'

10 head

+ echo '10 tail'

10 tail

+ (( i++ ))

+ '[' 11 -lt 12 ']'

+ echo '11 head'

11 head

+ echo '11 tail'

11 tail

+ (( i++ ))

+ '[' 12 -lt 12 ']'

Priyanka@LAPTOP-E2NR1CA4 MINGW64 ~/Desktop/practice problems

$