CS 100: Prelim 3 Solutions

April 27, 1999

7:30-9:00pm

____

(Print Name)

____

(Signature)

____

(Student ID)

Initial each of the following five(5) pages. For the sake of partial credit, you must show work and comment your solution fragments as appropriate. If you need space, write on the reverse side.

1. __________ (20 pts)

2. __________ (20 pts)

3. __________ (20 pts)

4. __________ (20 pts)

5. __________ (20 pts)

Total = __________ (100 pts)

1. (a) Indicate clearly what the output would be if the following `main` program is executed:

```
import java.io.*;
public class P1A{
   public static void f1(int[] a){
      a[0] = a[0]+10;
      a[1] = a[1]+20;
   }
   public static void main(String args[]){
       int[] a = {70, 30};
       int[] c = {70, 30};
       f1(c);
       System.out.println(a[0] + " " + a[1]);
       System.out.println(c[0] + " " + c[1]);
   }
}
```

Sol:

```
70 30
80 50
```

(b) Indicate clearly what the output would be if the following `main` program is executed:

```
import java.io.*;
public class P1B{
   public static int[] f2(int[] x){
      int[] y = new int[2];
      y[0] = x[0]+10;
      y[1] = x[1]+20;
      return y;
   }
   public static void main(String args[]){
      int[] x = {1,2};
      int[] y = x;
      x = f2(x);
      System.out.println(x[0] + " " + x[1]);
      System.out.println(y[0] + " " + y[1]);
   }
}
```

Sol:

```
11 22
1   2
```

2. **(a)** The following method is applied to a length-8 integer array. How many times is the message “blue” printed and how many times is the message “red” printed?

```
// Permutes the values in a so that a[0]<=a[1]<=...<=a[n-1]
// where n = a.length and n is a power of 2.
public static void MergeSort(int[] a){
   int n = a.length;
   if (n==1){
      System.out.println(“blue”);
      return;
   }
   else {
      int m = n/2;
      int[] aLeft  = new int[m];
      int[] aRight = new int[m];
      for(int i=0;i<m;i++){aLeft[i] =  a[i]; aRight[i] = a[i+m]; }

      MergeSort(aLeft);
      MergeSort(aRight);
      System.out.println(“red”);

      int iLeft=0;
      int iRight=0;
      for (int k=0;k<n;k++)
         if (iLeft==m)                            { a[k] = aRight[iRight]; iRight++;}
         else if (iRight==m)                      { a[k] = aLeft[iLeft];   iLeft++; }
         else if (aLeft[iLeft] <= aRight[iRight]) { a[k] = aLeft[iLeft];   iLeft++;}
         else                                     { a[k] = aRight[iRight]; iRight++;}
    }
  }
```

To receive full credit, you must justify your answer with an informal schmematic/picture that depicts the “divide and conquer” nature of the merge sort process. (Use the reverse side for your schenatic.)

Sol: blue 8 times and red 7 times.

(b) Indicate clearly the output that would be produced if the following fragment were executed:

```
int[] a = {10,20,30};
int[] b = new int[3];
for(int s=0;s<=1;s++){
   for(int j=0;j<3;j++){
      b[j] = a[(j+1)%3];
   }
   System.out.println(b[0] + “ “ + b[1] + “ “ + b[2]);
   a = b;
}
```

Sol:

20 30 10
30 10 30

3.Implement the following Java method so that it performs as specified.

```
// Assume that a[0]<= a[1] <= . . . <= a[n-1] where
// n is the length of a.
// Yields the number of distinct values in a.
public static int nDistinct(int[] a)
```

Thus, if the method is applied to the array

10	20	20	30	40

then it would return the value 4. Points will be deducted for grossly inefficient solutions.

Sol:

```
int n = a.length;
int s = 1;
for(int k=1;k<n;k++)
{
   if (a[k-1]<a[k])
   {
      s++;
   }
}
return s;
```

4. Complete the following method so that it performs as specified:

```
// Assume that a[0] < a[1] < . . . < a[n-1] where n is
// the length of a and assume L <= a[n-1].
// Yields a reference to an array whose values consist of all
// those values in a that are greater than or equal to L.
public static int[] subArray(int[] a, int L)
```

Thus, for the incoming array

10	20	30	40	50	60

and an L-value equal to 25, the method should return a reference to

30	40	50	60

Points will be deducted for solutions with loops that run unnecessarily long.

sol

```
int k=0;
while (a[k]<L)
   k++;
int m = a.length – k;
int[] b = new int[m];
for (int j=0;j<m;j++)
{
   b[j] = a[k];
   k++;
 }
 return b;
```

5. Assume the availability of the followiing Matlab function:

```
    function D = D98
% Yields a 2-by-365 matrix D.
% For i=1:365,
%    D(1,i) the high temperature for day i in 1998
%    D(2,i) the low  temperature for day i in 1998.
%
% Days are indexed from 1 (= Jan 1) to 365 (= December 31)
```

In the following, points will be deducted for solutions that require loops. In each case you should make effective use of the function `sum(x)` that returns the sum of the values in the vector `x`.

(a) Write a Matlab script that assigns to `n1` the number of days during 1998 whose low temperature is strictly bigger than 32.

```
D = D98;
n1 = sum(D(2,:)>32)
```

(b) Write a Matlab script that assigns to `n2` the number of days during 1998 for which the high temperature was was strictly 20 degrees above the low temperature.

```
D = D98;
n2 = sum((D(1,:)-D(2,:))>20)
```

(c) Write a Matlab script that assigns to `n3` the number of days from January 2 through December 30 for which the high temperature is strictly higher than the high temperature of the day before but strictly lower than the high temperature for the day after.

```
D = D98;
n3 = sum(D(1,1:363)<D(1,2:364) & D(1,2:364)<D(1,3:365))
```