Print Name

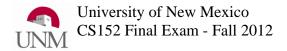
You may use one page of hand written notes (both sides) and a dictionary. No i-phones, calculators nor any other type of non-organic computer. Time Limit: 2 Hours

```
1) public class P1
 2) {
 3)
     public static void main(String[] args)
 4)
 5)
 6)
        int a = 2, b = 3, c = 5;
 7)
        String history="";
 8)
        if (a*b > c)
        { history += "Z";
 9)
          if (b < a) history += "Y";</pre>
10)
          else if (b < c) history += "X";</pre>
11)
          else if (c < b) history += "W";</pre>
12)
13)
         history += "V";
14)
        }
15)
       else if (b*c > a)
       { history += "T";
16)
          if (a > b) history += "S";
17)
18)
          else if (b > c) history += "R";
          else if (c > a) history += "Q";
19)
20)
          else if (c > b) history += "P";
21)
        }
22)
        else if ((a+b) \le c)
       { history += "N";
23)
24)
          if ((b - a) < c) history += "L";</pre>
          else if ((c - a) < b) history += "K";</pre>
25)
26)
          else if ((c - a) > b) history += "J";
27)
          else if ((c - a) == b) history += "H";
28)
        }
        history += "G";
29)
30)
        System.out.println(history);
31)
      }
32) }
```

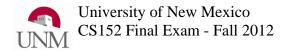
```
1) public class MethodsAndLogicExpressions
 2) {
     private static char test(int x1, int y1, int x2, int y2)
 3)
 4)
      { if ((x1 < 99) \&\& (x2 < 99)) return 'W';
        if ((x1 > 150) && (x2 > 150)) return 'E';
 5)
 6)
        if ((y1 < 25) \&\& (y2 < 25)) return 'N';
 7)
        if ((y1 > 75) \&\& (y2 > 75)) return 'S';
 8)
        if ((x1 < 99) \mid | (x1 > 150) \mid | (y1 < 25) \mid | (y1 > 75))
 9)
10)
          if ((x2 < 99) \mid | (x2 > 150) \mid | (y2 < 25) \mid | (y2 > 75))
11)
          { return 'O';
12)
13)
          else return 'X';
14)
        }
15)
        else
16)
          if ((x2 < 99) \mid | (x2 > 150) \mid | (y2 < 25) \mid | (y2 > 75))
17)
18)
          { return 'Y';
19)
20)
          else return 'I';
21)
        }
22)
      }
23)
24)
25)
     public static void main(String[] args)
26)
     {
27)
        System.out.println("1" + test(0, 0, 10, 10));
        System.out.println("2" + test(120,50, 130,70));
28)
29)
      }
30) }
```

3) What would be the output of MethodsAndLogicExpressions if lines 27 and 28 are replaced with:

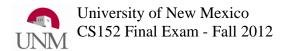
```
27)     System.out.println("3" + test( 0,50, 200,50));
28)     System.out.println("4" + test(120,50, 200,50));
```



```
1)
    public class P4
 2)
3)
       public static int foo(int a, int b)
 4)
 5)
         if (a > b)
         { int c=a; a=b; b=c;
 6)
7)
 8)
         for (int i=a; i>1; i--)
 9)
10)
           System.out.print(i + " "); // *** OUTPUT ****
           if ((a % i) == (b % i))
11)
12)
13)
             return i;
14)
           }
15)
         }
16)
         return 1;
17)
18)
       public static void main(String[] args)
19)
         System.out.println("==>" + foo(5, 8)); // *** OUTPUT ***
20)
         System.out.println("==>" + foo(9, 5)); // *** OUTPUT ***
21)
         System.out.println("==>" + foo(5, 9)); // *** OUTPUT ***
22)
23)
       }
24)
     }
```



```
1)
     public class LoopNest
 2)
 3)
       public static void main(String[] args)
 4)
 5)
          String[] animation =
          { "Troll", "Spider", "Rat", "Zombie"
 6)
 7)
          };
 8)
          String[] type = {"jpg", "png", "gif"};
 9)
10)
          for (int a=1; a<animation.length; a++)</pre>
11)
            for (int b=0; b<type.length; b++)</pre>
12)
13)
              for (int c=2; c<4; c++)</pre>
14)
15)
                String str = animation[a] + c + '.' + type[b];
16)
17)
18)
                System.out.println(str); // *** OUTPUT ***
19)
              }
20)
            }
21)
          }
22)
        }
23)
      }
```



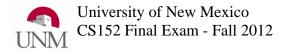
```
1) public class ArrayRangeScaleAndShift
 2) {
 3)
      public static void main(String[] args)
 4)
 5)
        int[] x = {2, -3, -2, 1, 3, 0};
        int min = -3;
 6)
 7)
        int max = 3;
        int min2 = 2;
 8)
        int max2 = 17;
 9)
10)
11)
        for (int i=0; i<x.length; i++)</pre>
12)
          int y = min2 + (x[i]-min) * (max2-min2)/(max-min);
13)
          int z = min2 + (x[i]-min) * ((max2-min2)/(max-min));
14)
15)
16)
          System.out.println(
17)
            x[" + i + "] = x[i] + "--> y = y + y + z;
18)
       }
19)
      }
20) }
```

7) This Java program uses Picture class developed in CS-152. The program compiles, runs and displays a JFrame window with a black background. Sketch an image of what is drawn within the JFrame. This is a sketch. You need to capture the approximate shapes and patterns. You do not need to reproduce the exact measurements, nor the exact number of lines.

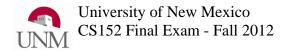
```
1) import java.awt.Graphics;
 2) import java.awt.Color;
 3)
 4) public class Test
 5) {
     public static void main(String[] args)
 6)
 7)
 8)
       Picture myPic = new Picture (201, 201);
 9)
       Graphics canvas = myPic.getOffScreenGraphics();
10)
       canvas.setColor(Color.GREEN);
11)
        for (int a=0; a<100; a+=20)</pre>
12)
13)
          int b = 100+a;
14)
          int c = 100-a;
15)
         canvas.drawLine(b,
                               2*a, b , 2*a+10);
          canvas.drawLine(b-5, 2*a+5, b+5, 2*a+5);
16)
17)
18)
          canvas.drawLine(c,
                               2*a,
                                      C,
          canvas.drawLine(c-5, 2*a+5, c+5, 2*a+5);
19)
20)
21)
       myPic.repaint();
22)
     }
23) }
```

8) This Java program uses Picture class developed in CS-152. The program compiles, runs and displays a JFrame window with a black background. Sketch an image of what is drawn within the JFrame. This is a sketch. You need to capture the approximate shapes and patterns. You do not need to reproduce the exact measurements, nor the exact number of lines.

```
1) import java.awt.Graphics;
 2) import java.awt.Color;
 3) public class Test
 4) {
 5)
      public static void main(String[] args)
 6)
 7)
        Picture myPic = new Picture(425, 201);
 8)
        Graphics canvas = myPic.getOffScreenGraphics();
 9)
        canvas.setColor(Color.RED);
10)
        for (int n=0; n<16; n++)</pre>
11)
           int a = 20 + ((n*50) \% 200);
12)
13)
          int x = n*25+20;
          for (int i=-2; i<=2; i++)</pre>
14)
15)
16)
             canvas.drawLine(x, a, x-i*5, a-10);
17)
18)
          canvas.drawLine(x, a, x, a+10);
19)
        }
20)
      }
21) }
```



```
1)
     public class Histogram
 2)
 3)
       public static void main(String[] args)
 4)
        {
 5)
          char[] grade =
          { 'C', 'B', 'C', 'C', 'B', 'C', 'C', 'A', 'D', 'B'
 6)
 7)
          int[] hist = new int[5];
 8)
 9)
10)
          for (int i = 0; i < grade.length; i++)</pre>
11)
            int numGrade = grade[i] - 'A';
12)
13)
            hist[numGrade]++;
14)
            System.out.println(i + ":"
              + " hist[" + numGrade + "]=" + hist[numGrade]);
15)
16)
          }
17)
        }
18)
     }
```



```
1)
     public class MagicSquare
2)
      public static boolean isMagic(String str)
3)
4)
5)
        System.out.println("isMagic("+str+"):"); //<-----</pre>
6)
7)
         int sum = 0;
         if (str.length() != 3) return false;
8)
         for (int i=0; i<str.length(); i++)</pre>
9)
10)
         {
          char c = str.charAt(i);
11)
12)
13)
          if (!Character.isDigit(c)) return false;
14)
15)
          sum = sum + Character.getNumericValue(c);
16)
          if (sum > 15) return false;
          System.out.println(" "+c+ ", sum=" +sum); //<-----</pre>
17)
18)
19)
        }
20)
        if (sum == 15) return true;
21)
        return false;
22)
       }
23)
24)
      public static void main(String[] args)
25)
         System.out.println(isMagic("15"));
                                             //<----
26)
                                             //<----
27)
         System.out.println(isMagic("x39"));
         System.out.println(isMagic("239"));
                                            //<----
28)
                                             //<----
29)
        System.out.println(isMagic("987"));
        30)
31)
       }
32)
     }
```