CECS 524 Unit 6 Assignment

Name: Spuritha Mudireddy

CSULB ID: 030743269

Using the programming language of your choice, implement the expression for computing array addresses for 2-dimensional arrays of any element size and any arbitrary lower and upper bounds. This is the expression

location(a[i, j]) = address of a[row_lb, col_lb] - (((row_lb * n)+ col_lb) * element_size) + (((i * n) + j) * element_size)

<u>Code:</u>

```
package unit6;

public class ComputeAddress {

	public static void main(String[] args) {
		calcAddress(1200, 0, 0, 2, 2, 1);
		calcAddress(100, 1,1,2,2,2);
		calcAddress(100, 2, 3, 4,5,4);
		calcAddress(100, -1, -1, 1, 2, 8);

	}

	public static void calcAddress(int base,int  rowlb,int   collb,int
rowub,int   colub,int   elementsize)
	{
		int a[]=new int[] {base,rowlb,collb,rowub,colub,elementsize};
		//System.out.println(a.length);
		System.out.println("For array
a["+rowlb+":"+rowub+","+collb+":"+colub+"] with element size "+elementsize);

		for(int i=rowlb;i<=rowub;i++)
		{
			for(int j=collb;j<=colub;j++)
```

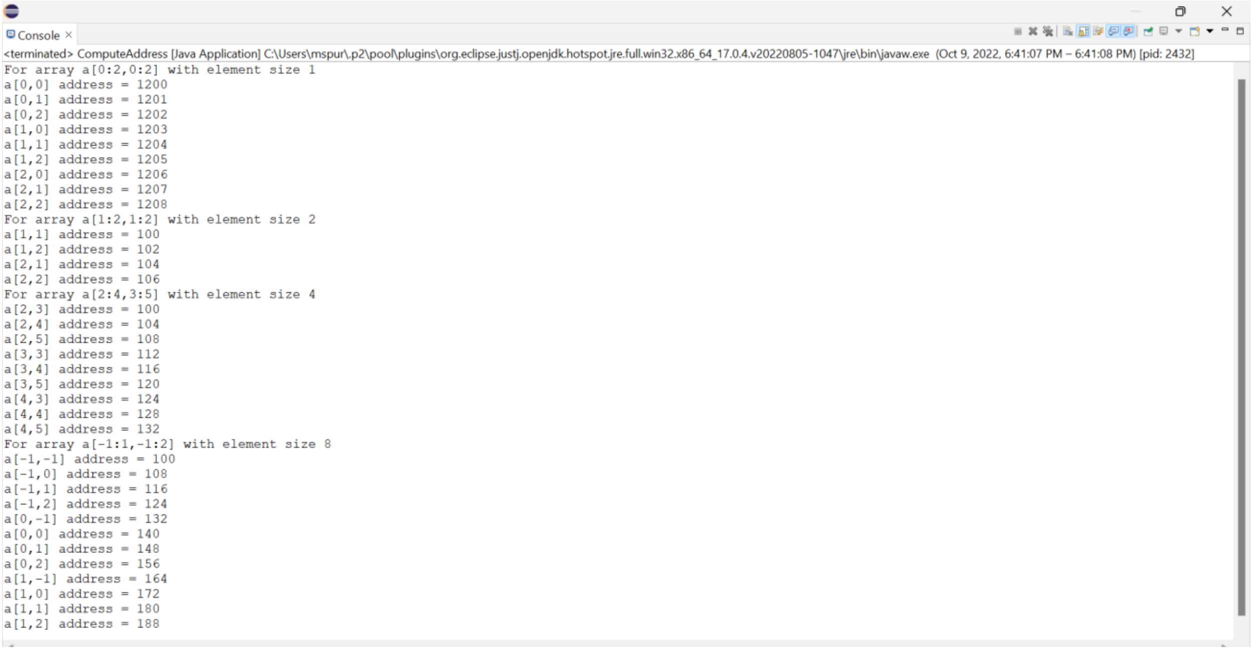
```
            {
                int n=colub-collb+1;
                int loc=base - (((rowlb * n)
                        + collb) * elementsize) + (((i * n) + j)
* elementsize);
                System.out.println("a["+i+","+j+"] address = "+loc);

            }

        }
    }

}
```

Output:



<terminated> ComputeAddress [Java Application] C:\Users\mspur\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220805-1047\jre\bin\javaw.exe (Oct 9, 2022, 6:41:07 PM – 6:41:08 PM) [pid: 2432]

```
For array a[0:2,0:2] with element size 1
a[0,0] address = 1200
a[0,1] address = 1201
a[0,2] address = 1202
a[1,0] address = 1203
a[1,1] address = 1204
a[1,2] address = 1205
a[2,0] address = 1206
a[2,1] address = 1207
a[2,2] address = 1208
For array a[1:2,1:2] with element size 2
a[1,1] address = 100
a[1,2] address = 102
a[2,1] address = 104
a[2,2] address = 106
For array a[2:4,3:5] with element size 4
a[2,3] address = 100
a[2,4] address = 104
a[2,5] address = 108
a[3,3] address = 112
a[3,4] address = 116
a[3,5] address = 120
a[4,3] address = 124
a[4,4] address = 128
a[4,5] address = 132
For array a[-1:1,-1:2] with element size 8
a[-1,-1] address = 100
a[-1,0] address = 108
a[-1,1] address = 116
a[-1,2] address = 124
a[0,-1] address = 132
a[0,0] address = 140
a[0,1] address = 148
a[0,2] address = 156
a[1,-1] address = 164
a[1,0] address = 172
a[1,1] address = 180
a[1,2] address = 188
```