

Game Engineering 2D
Midterm 1 problems
Fall 2010

1. Give the output of the following C# Program:

```
struct s1
{
    int x;
    int y;
}
class C1
{
    s1 str;
}
struct S2
{
    s1 str1;
    C1 cls;
}

class Program
{

    void twiddle1(S2 str)
    {
        str.str1.x = 1;
        str.str1.y = 2;
        str.cls.str.x = 3;
        str.cls.str.y = 4;
    }

    void twiddle2(S2 var str)
    {
        str.str1.x = 5;
        str.str1.y = 6;
        str.cls.str.x = 7;
        str.cls.str.y = 8;
    }

    public static void main(string[] args)
    {
        S2 str = new S2();
    }
}
```

```

        str.str1 = new S1();
        str.cls = new C1();
        str.cls.str = new S1();
        twiddle1(str);
        Console.WriteLine(str.str1.x);
        Console.WriteLine(str.str1.y);
        Console.WriteLine(str.cls.str.x);
        Console.WriteLine(str.cls.str.y);
        twiddle2(str);
        Console.WriteLine(str.str1.x);
        Console.WriteLine(str.str1.y);
        Console.WriteLine(str.cls.str.x);
        Console.WriteLine(str.cls.str.y);
    }
}

```

Output:

```

0
0
3
4
5
6
7
8

```

2. The following function is supposed to insert an element into a sorted linked list. It is, however, buggy. Fix the errors.

```

class ListElem
{
    public int data;
    public ListElem next;
    public ListElem prev;
}

```

```

class List
{
    ListElem first;
    ListElem last;

    void Insert(int data)
    {

```

```

ListElem e = new ListElem();
if (first == null)
{
    first = new ListElem();
    last = new ListElem();
    first.data = data;
}
ListElem tmp = first;
while (tmp.data > elem)
{
    tmp = tmp.next;
}
e.prev = tmp.prev;
e.next = tmp;
tmp.prev = e;
tmp.prev.next = e;
}
}

```

So many errors ...

- In the special case for inserting into an empty list, last and first should be assigned to the same value.
- In the special case for inserting into an empty list, you should not *also* fall through to the general case for inserting into a non-empty list
- The while loop will fall off the end of the list when inserting an element at the end of the list
- Need to set the first / last pointers when inserting at the front / end of the list

There is more than one way to fix these bugs, here is one:

```

class ListElem
{
    public int data;
    public ListElem next;
    public ListElem prev;
}

class List
{
    ListElem first;
    ListElem last;

    void Insert(int data)

```

```

{
    ListElem e = new ListElem();
    e.data = data;
    if (first == null)
    {
        first = e;
        last = e;
    }
    else
    {
        if (first.data <= data)
        {
            e.next = first;
            first.prev = e;
            first = e;
        }
        else
        {
            ListElem tmp = first;
            while (tmp.next != null && tmp.next.data < data)
            {
                tmp = tmp.next;
            }
            e.next = tmp.next;
            e.prev = tmp;
            tmp.next = e;
            if (e.next == null)
            {
                last = e;
            }
            else
            {
                e.next.prev = e;
            }
        }
    }
}
}
}

```

3. A robot has an orientation defined by the matrix M_r and position p_r , in global space. The robot's hand has an orientation and position defined by M_h and p_h *in the local space of the robot*. The robot is grasping a gun, which is at orientation and position M_g and p_g *in the local space of the hand of the robot*. Give the position and orientation of the gun in Global Space.

Position of gun in world space: $(p_g * M_h + p_h) * M_r + p_r$

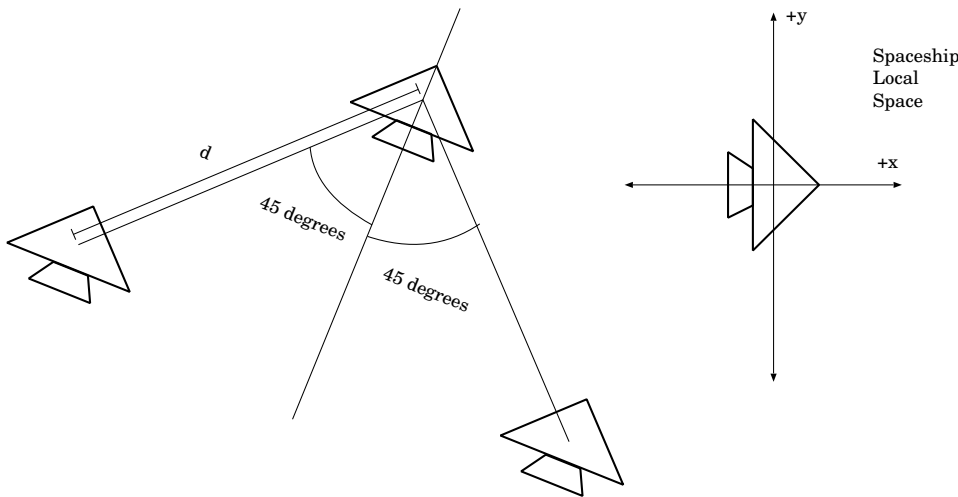
Orientation of gun in world space: $M_g * M_h * M_r$

4. A spaceship s has an orientation defined by the following rotational matrix:

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

and position $p = [x, y]$.

The spaceship points down the x axis in its local space. The ship is flying in formation with two ships s_2 and s_3 that are at a distance d_1 away, each at a 45 degree angle from the original ship:



Give the orientations (as rotational matrices) and positions of s_2 and s_3 , as a function of a, b, c, d, x, y .

First, we will look at the orientations, since those are easiest: The orientation of the ships are the same as the orientation of the original ship:

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The vectors from the lead ship to each of the following ships (in the local space of the lead ship) are $d_1 * [\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}]$ and $d_1 * [\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}]$. In global space, those vectors are

$$d_1 * \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = d_1 * \begin{bmatrix} \frac{a-c}{\sqrt{2}} & \frac{b-d}{\sqrt{2}} \end{bmatrix}$$

and

$$d_1 * \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = d_1 * \begin{bmatrix} \frac{-a-c}{\sqrt{2}} & \frac{-b-d}{\sqrt{2}} \end{bmatrix}$$

Add each of these vectors to the position of the lead ship in global space, and you get:

$$[x, y] + d_1 * \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \left[x + \frac{d_1(a - c)}{\sqrt{2}}, y + \frac{d_1(b - d)}{\sqrt{2}} \right]$$

and

$$[x, y] + d_1 * \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \left[x + \frac{d_1(-a - c)}{\sqrt{2}}, y + \frac{d_1(-b - d)}{\sqrt{2}} \right]$$