Homework

Problem 1.

The following problem illustrates the way memory aliasing can cause unexpected program behavior. Consider the following procedure to swap two values:

```
/* move value x at xp to y at yp */
int move(int *xp, int *yp)
{
*yp = *xp; /* copy */
free(xp);
return *yp;
}
```

If this procedure is called with xp equal to yp, what effect will it have?

Problem 2.

Consider the following functions:

```
int min(int x, int y) { return x < y ? x : y; }
int max(int x, int y) { return x < y ? y : x; }
void incr(int *xp, int v) { *xp += v; }
int square(int x) { return x*x; }
```

The following three code fragments call these functions:

A. For(i = min(x,y); i< max(x,y);incr(&x,i))
   m+=square(i);
B. For(i = max(x,y)-1; i>=min(x,y); incr(&I,1))
   m+=square(i);
C. Int low=min(x,y);
   Int high=max(x,y);
   For(i=low, i<high;i=incr(&i,1))
   m+=square(i);

Assume x equals 3 and y equals 15. Fill in the following table indicating the number of times each of the four functions is called in code fragments A–C:

	min	max	incr	square
A				
B				
C				

Problem 3.

The marketing department at your company has promised your customers that the next software release will show a 2× performance improvement. You have been assigned the task of delivering on that promise. You have determined that only 80% of the system can be improved. How much (i.e., what value of k) would you need to improve this part to meet the overall performance target?

Problem 4.

Suppose you are given the task of improving the performance of a program consisting of three parts. Part A requires 20% of the overall run time, part B requires 30%, and part C requires 50%. You determine that for $1000 you could either speed up part B by a factor of 3.0 or part C by a factor of 1.5. Which choice would maximize performance?