Midterm #3

27 April, 1994

This is an open books, open notes exam to be turned in by 10:50 AM.

Name: _____

Problem 1. (20 points)

Write a C procedure that takes a pointer to an integer as an argument and multiplies the integer by two. Call the procedure `DoubleIt` and start with the following outline:

```
DoubleIt(int *pI)
  {
   /* You add stuff in here */


  } /* DoubleIt */
```

Suppose your program contains an array `A` of 15 integers. Write a loop that will double each element of `A` by calling `DoubleIt` while passing, one at and time, each element of `A`. Start with the following declarations:

```
int A[15], I ;
```

Problem 2. (20 points)

Consider each of the following four loops

```
/* Loop A */
for (i=0; i<n; ++i)
   for (j=i; j<n; ++j)
      s += A[i][j] ;
```

```
/* Loop B */
for (i=0; i<10; ++i)
   for (j=1; j<n; ++j)
      s += A[i][i] ;
```

```
/* Loop C */
for (i=0; i<1000; ++i)
   for (j=1; j<i; ++j)
      s += A[i][j] ;
```

```
/* Loop D */
for (i=n; i>0; --i)
   for (j=1; j<n; j *= 2)
      s += A[i][j]
```

In big-O notation, the running times of these algorithms are O(1), O(n), O(n log n), and O(n^2). Match each algorithm with its running time.

Problem 3. (20 points)

Consider the following binary search algorithm for seeking an item `k` within an array `A`.

```
low = 0 ;
high = MAXSIZE - 1 ;
while (low<=high) {
   mid = (low+high) / 2 ;
   if (A[mid]<k)
      low = mid+1;
   else
      high = mid ;
}
```

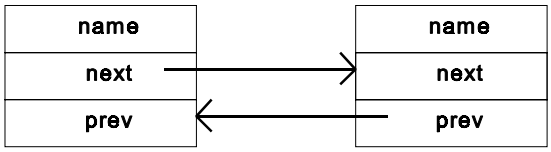
If `A` is of size seven and contains the elements

14 20 23 27 33 40 48

and `k` is 21, what are the values of `low` and `high` at the beginning of each iteration of the binary search loop?

The two remaining problems are concerned with doubly linked lists. Use the following C data structure definition in your answers.

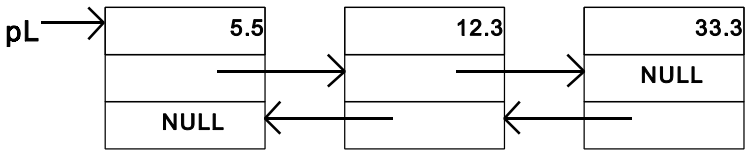
```
typedef struct dl_node_tag {
   float name ;
   struct dl_node *next ;
   struct dl_node *prev ;
} DL_Node_type ;
```



PS: This *should* look familiar.

Problem 4. (20 points)

Starting with the variable `pL` pointing to the leftmost node of the doubly linked list on the right, show the result of executing the following five C statements in order.




```
pL->prev = pL->next ;              /* 1 */
pL->next->next = pL->next ;        /* 2 *
pL->next = pL->prev->next ;        /* 3 */
pL->name = 1492.0 ;                /* 4 */
pL = pL->prev ;                    /* 5 */
```

You may find it useful to use the numbers in the comments to document the changes.

Problem 5. (20 points)

Again, suppose you have two pointers, `pLeft` and `pRight`, declared as follows:

```
DL_Node_type *pLeft, *pRight;
```

that point to the left and right ends of a doubly linked list.

This time write a few lines of C to perform a ``circular left shift'' on the list. That is move the present left end of the list to follow the present right end of the list. Set `pLeft` and `pRight` to point to the new ends of the list.