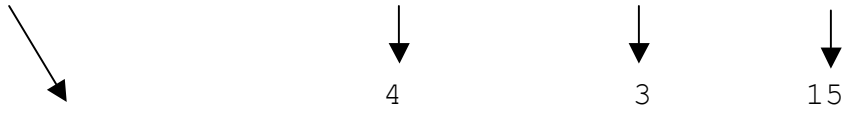
Final Exam Review

1. 2D Arrays What is the value returned from **`guess`**?

20	5	10
30	10	40
10	20	50
15	3	20




```
int guess( int table[][3], int rows, int cols, int limit )
{
   int r;
   int c;
   int res;

   res = 0;
   for( c = 0; c < cols; c++ )
      for( r = 0; r < rows; r++ )
           if( table[r][c] < limit )
               res += table[r][c];
   return res;
}
```

2. Pointers Draw the memory map for the following program fragment. Predict the output.

```
   char ch1 = 'O';
   char ch2 = 'B';
   char *q;
   char *r;
   char *p;



   q = &ch2;
   r = &ch1;
   p = q;
   cout << *q << " " << *r << " " << *p;
   p = r;
   cout << *q << " " << *r << " " << *p;
```

3. Pointers and Arrays Predict the output.

```
// Local Declarations
   int list[100] = {10, 11, 30, 20, 15, 40, 25, 0};
   int *ptr;
   int  num;

// Statements

   for( ptr = list; *ptr != 0; ptr++ )
   {
      *ptr = *ptr * 2 ;
       cout << *ptr << " ";
   }
```

4. Pointers and Structures What's wrong with the following program fragment? How would you correct it?

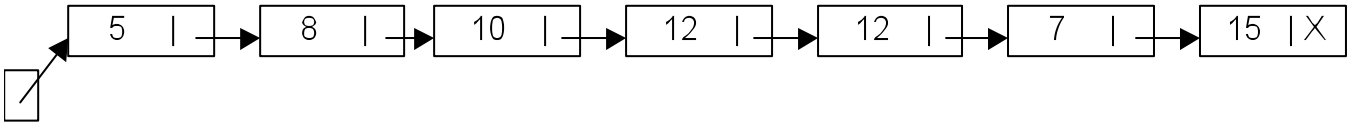
```
struct Exam{
   int  key;
   char ch;
};
void getData( Exam *x );
int main( void )
{
   Exam *a;

   getData( a );
   . . .
   return 0;
}
/* ======================== */
void getData( Exam *x )
{
   cin << x->key ;
   cin << x->ch;
}
```

5. What is wrong with the following program fragment? Circle and correct the error(s).

```
ListNode *pN;
int   count = 0;
// . . .
for( pN = head; pN != NULL; pN++ )
{
   if( pN.data > 0 )
       cout << pN.data << " ";
   else
       count++;
}
```

6. Linked Lists What is the value returned from the function guess if the linked list is:




```
int NumberList::guess( )
{
   ListNode *pN = head;
   int num = -1;

   if( pN != NULL )
   {
      num   = pN->data;
      pN = pN->next;
      while( pN != NULL && num <= pN->data )
      {
        num = pN->data;
        pN = pN->next;
      }
   }
   return num;
}
```

7. You have a list of books that needs to be updated. The **BOOK** structure contains two fields: **qty**, quantity, an integer value, and **title**, a string of size 30. Write a function named **update** that searches the unsorted array to find the given title. If found, add 1 to qty; otherwise, if there is room, append the new title at the end of the array and set qty to 1. This function has three parameters: the array, its length, and the title.

8. Write the definition of a swap function given its calling statements below:

```
success = swap(3, 5);
success = swap(5, 3);
```

The purpose of the function is to swap node number 3 and node number 5 (any two nodes) in a linked list, if possible. Note: update the links rather than copying data.