Problems for Midterm 1 Review
During the test you will be allowed to access the documentation of Point Class and Bag ADT.

1. Consider the following program.
   ```cpp
   int main()
   {
       ifstream is ("input_points.txt");
       Point P[5];
       Point M(0,0);
       int i,k,m,n;
       Point Q;

       for(i=0;i<5;i++)
           is >> P[i];

       for(k=4;k>0;k--)
       {
           M.set(0,0);
           for(m=0;m<=k;m++)
           {
               if(P[m].get_y()>M.get_y())
               {
                   M=P[m];
                   n=m;
               }
           }
           Q=P[n];
           P[n]=P[k];
           P[k]=Q;
           for(i=0;i<5;i++)
           {
               cout << P[i] << " ";
           }
           cout << endl;
       }
   return 0;
   }
   ```

   What output do we get if for input we take:
   
   1 5
   3 7
   4 3
   6 8
   2 4
2. The following program should read points from an input file and compute the sum of those points after their coordinates have been raised to the third power. Fill in the gaps in the main program and in the implementation of the member function power_3.

```cpp
void Point::power_3()
{
    _x=…
    _y=…
}

int main()
{
    ifstream is ("input_points.txt");
    Point P;
    Point Q;

    while( is >> P )
    {
        P.power_3();
        Q=…;
    }
    cout << Q << endl;
    return 0;
}
```

3. The following program should prompt the user for a number, allocate an integer type array of the size entered by the user, read data into this array and print out the largest element of the array. Fill in the blanks.

```cpp
int main()
{
    int n,i,max;
    cout << "Enter a number: ";
    cin >> n;

    …;
    …;
    for(i=0;i<n;i++)
    {
        cin >> A[i];
    }
    max=A[0];
    for(i=1;i<n;i++)
    {
        if(A[i]>max)
        {
            …;
        }
    }
    cout << max << endl;;
```
4. Consider the program.
#include <iostream>
#include <fstream>
using namespace std;

struct Node
{
    typedef int Item;
    Item data;
    Node *link;
};

void list_insert_head(Node* & head_ptr, Node::Item m)
{
    Node *cursor;
    cursor = new Node;
    cursor->data = m;
    cursor->link = head_ptr;
    head_ptr = cursor;
}

int main()
{
    Node *head_ptr = NULL;
    Node *cursor = NULL;
    Node::Item m;
    ifstream is("input.txt");

    if(is >> m)
    {
        head_ptr = new Node;
        head_ptr->data = m;
        cursor = head_ptr;
    }

    while(is >> m)
    {
        cursor->link = new Node;
        cursor = cursor->link;
        cursor->data = m;
    }

    if(cursor != NULL)
    {
        cursor->link = NULL;
    }
}
Node *new_head_ptr=NULL;
for(cursor=head_ptr;cursor!=NULL;cursor=cursor->link)
    list_insert_head(new_head_ptr,cursor->data);

for(cursor=new_head_ptr;cursor!=NULL;cursor=cursor->link)
{
    cout << cursor->data << endl;
}
return 0;

(i) What output do we get if for input we take:
11 12 13 14 15
(ii) What is the output if the header of the output loop is changed to:
for(cursor=head_ptr;cursor!=NULL;cursor=cursor->link)
(iii) What is the output if we change the header of the list_insert_head function to:
void list_insert_head(Node* head_ptr,Node::Item m) (the header of the output
loop goes back to its original form)

5. A linked list with integer data is given. Write a C++ function, which will return
the address of the node which contains the largest data field.

6. What is the output of the following program?
int main()
{
    Bag D;
    int i;
    for(i=0;D.occurrences(10)==0;i++)
        D.insert(i);
    D=D+D+D;
    cout << D.occurrences(1) << endl;
    cout << D.occurrences(2) << endl;
    cout << D.occurrences(10) << endl;
    cout << D.occurrences(11) << endl;
    cout << D.size()<< endl;
return 0;
}