Q1-13 (should be clear from graded exam paper)

Q14
```
IntSet IntSet::symmDiff(const IntSet& otherIntSet) const
{
   IntSet answer(used + otherIntSet.used); // has capacity good for worst case
   int i, j;
   for (i = 0; i < used; ++i)
   {
      for (j = 0; j < otherIntSet.used; ++j)
         if (otherIntSet.data[j] == data[i]) break;
      if (j == otherIntSet.used) answer.data[answer.used++] = data[i];
   }
   for (i = 0; i < otherIntSet.used; ++i)
   {
      for (j = 0; j < used; ++j)
         if (data[j] == otherIntSet.data[i]) break;
      if (j == used) answer.data[answer.used++] = otherIntSet.data[i];
   }
   return answer;
}
```

Insert `IntSet symmDiff(const IntSet& otherIntSet) const;` into the `public` section of `IntSet`'s class declaration, most logically as a new member of the block comprising of accessors.

Q15
```
IntSet mySymmDiff(const IntSet& is1, const IntSet& is2)
{  // it's possible to replace following 2 statements with just 1 statement
   IntSet union12 = is1.unionWith(is2), intersect12 = is1.intersect(is2);
   return union12.subtract(intersect12);
}
```