Data Structures, Algorithms, & Applications in C++
Chapter 1, Exercise 5

We may represent a subset of n elements by the one-dimensional array x[1:n], where x[j] is one if element j is included in the subset and x[j] is zero if element j is not included in the subset.

To output the subsets recursively, we define a function Subsets(int i) which outputs all x[1:n] with preset values for x[1:i-1] and x[i:n] taking on all possible 0 and 1 values. The invocation Subsets(1) will output all subsets.

The code is given below and in the files rsubset.*. The code assumes that n and x are global variables.

```
void Subsets(int i)

{// Output all subsets of x[1:n].

 // Only x[i:n] to be changed.

   if (i == n) {// x[n] can be 0 or 1

                // output subset without element n

                x[n] = 0;

                for (int j = 1; j <= n; j++)

                   cout << x[j] << " ";

                cout << endl;



                // output subset with element n

                x[n] = 1;
```

```
                for (int j = 1; j <= n; j++)
                   cout << x[j] << " ";
                cout << endl;
                return;
                }

    // leave element i out
    x[i] = 0;
    // generate all subsets with i out
    Subsets(i+1);

    // put element i into subset
    x[i] = 1;
    // generate all subsets with i included
    Subsets(i+1);
}
```

The above code may be modified if we are to ouptut element identifiers for the selected elements rather than 0/1 vectors.