CSE 16
Lab Assignment 2

The goal of this assignment is to determine all 5-element subsets of a 9-element set. As we know, there are $2^9 = 512$ subsets of a 9-element set. One way to proceed is to generate all of these subsets, then filter out those that do not contain exactly 5 elements. There are of course other methods to solve this problem, and you are at liberty to use any technique you like.

This is similar to what we did in lab1, where we generated all bit strings of length 10, then filtered out those that did not satisfy a certain logical condition. In fact, we have discussed in class a bijection between the set of bit strings of length n and the subsets of an n-element set. We illustrate here with $n = 5$, and the set $\{1, 2, 3, 4, 5\}$.

Bit Strings		Subsets
0 0 0 0 0	⟷	{}
0 0 0 0 1	⟷	{5}
0 0 0 1 0	⟷	{4}
0 0 0 1 1	⟷	{4, 5}
0 0 1 0 0	⟷	{3}
0 0 1 0 1	⟷	{3, 5}
0 0 1 1 0	⟷	{3, 4}
0 0 1 1 1	⟷	{3, 4, 5}
0 1 0 0 0	⟷	{2}
0 1 0 0 1	⟷	{2, 5}
0 1 0 1 0	⟷	{2, 4}
0 1 0 1 1	⟷	{2, 4, 5}
0 1 1 0 0	⟷	{2, 3}
0 1 1 0 1	⟷	{2, 3, 5}
0 1 1 1 0	⟷	{2, 3, 4}
0 1 1 1 1	⟷	{2, 3, 4, 5}
1 0 0 0 0	⟷	{1}
1 0 0 0 1	⟷	{1, 5}
1 0 0 1 0	⟷	{1, 4}
1 0 0 1 1	⟷	{1, 4, 5}
1 0 1 0 0	⟷	{1, 3}
1 0 1 0 1	⟷	{1, 3, 5}
1 0 1 1 0	⟷	{1, 3, 4}
1 0 1 1 1	⟷	{1, 3, 4, 5}
1 1 0 0 0	⟷	{1, 2}
1 1 0 0 1	⟷	{1, 2, 5}
1 1 0 1 0	⟷	{1, 2, 4}
1 1 0 1 1	⟷	{1, 2, 4, 5}
1 1 1 0 0	⟷	{1, 2, 3}
1 1 1 0 1	⟷	{1, 2, 3, 5}
1 1 1 1 0	⟷	{1, 2, 3, 4}
1 1 1 1 1	⟷	{1, 2, 3, 4, 5}

Formally, the correspondance from subsets $A \subseteq \{1, 2, \dots, n\}$ to bit strings $b = b_1 b_2 \cdots b_n$ of length n is

$$b_i = \begin{cases} 1 & \text{if } i \in A \\ 0 & \text{if } i \notin A \end{cases}$$

One checks that the inverse of this map, taking bit strings to subsets, is

$$A = \{\, i \mid 1 \le i \le n \text{ and } b_i = 1\}$$

This check verifies that the maping is invertible, and hence a bijection. Notice that the above map also imposes a standard order on the subsets of an n-element set. By interpreting each bit string as a non-negative integer written in binary notation, we can arrange the bit strings increasing order, then arrange the subsets in corresponding order. This is illustrated above in the case $n = 5$. If we wish to list all of the 3-element subsets of $\{1, 2, 3, 4, 5\}$ in order, we have

Bit Strings		Subsets
0 0 1 1 1	⟷	{3, 4, 5}
0 1 0 1 1	⟷	{2, 4, 5}
0 1 1 0 1	⟷	{2, 3, 5}
0 1 1 1 0	⟷	{2, 3, 4}
1 0 0 1 1	⟷	{1, 4, 5}
1 0 1 0 1	⟷	{1, 3, 5}
1 0 1 1 0	⟷	{1, 3, 4}
1 1 0 0 1	⟷	{1, 2, 5}
1 1 0 1 0	⟷	{1, 2, 4}
1 1 1 0 0	⟷	{1, 2, 3}

Let m be the number of 5-element subsets of the 9 element set $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. An output file for this project will consist of a short paragraph describing your method of solving the problem, then a blank line, then m lines giving a list of your subsets in order, followed by "`count = m`" on a single line by itself. Illustrating again here with the 3-element subsets of $\{1, 2, 3, 4, 5\}$, the listing would appear as

```
{3, 4, 5}
{2, 4, 5}
{2, 3, 5}
{2, 3, 4}
{1, 4, 5}
{1, 3, 5}
{1, 3, 4}
{1, 2, 5}
{1, 2, 4}
{1, 2, 3}
count = 10
```

Notice that a subset appears within curly braces { }, its elements apear in increasing order, and each element other than the last is followed by a comma then a space.

Place everything (paragraph, blank line, listing, count) in a text file called **`lab2.txt`** and submit it to Gradescope by the due date. Depending on how you did lab1, this project may be just an easy extension of it. All you have to do is to figure out how to generate the right bit strings, then print out the set corresponding to each such string. Even so, you should not wait until the last minute to begin. Start early and ask questions when necessary.