Set

A set is a well-defined collection of distinct objects. The objects that make up a set (also known as the set's elements or members) can be anything: numbers, people, letters of the alphabet, other sets, and so on.

In this viva session, you are assigned to create a generic *Set* class using Arraylist implementation.

Q1 [2 marks]:

Create a generic class called *Set* with following attributes and methods:

- A generic type Arraylist attribute that store elements of set
- A constructor, *Set(array of elements)* that initialize Arraylist attribute
- A method, *union(set)* that return a new set contains all elements in *this* set OR in another set
- A method, *intersection(set)* that return a new set contains all elements in *this* set AND in another set
- A method, *complement(set)* that return a new set contains all elements in *this* set BUT NOT in another set
- A method, *isSubset(set)* that return a boolean whether *this* set is subset of another set
- A method, *cardinality()* that return number of elements in *this* set
- A method, *toString()* that return string that shows all elements in *this* set

Some recaps:

- Let Set A = {1, 2, 2, 3} and Set B = {1, 3, 5, 8}
- Set does not contain duplicate element. Thus, Set A = {1, 2, 2, 3} = {1, 2, 3}.
- Set A union Set B should yield {1, 2, 3, 5, 8}.
- Set A intersect Set B should yield {1, 3}.
- Set A complement Set B should yield {2}.
- Set A is a subset of Set B if and only if all elements in Set A are also in Set B but Set A ≠ Set B (proper subset).

Create a Tester class and test your program with following output.

```
Set A: {1, 2, 3, 4, 5, 6}
Set B: {1, 2, 5, 9, 13}
Set C: {1, 2, 5, 9, 13, 14}
Set D: {1, 2, 5, 9, 13}
Set A union Set B: {1, 2, 3, 4, 5, 6, 9, 13}
Set A intersect Set B: {1, 2, 5}
Set A complement Set B: {3, 4, 6}
Is Set B subset of Set C? true
Is Set B subset of Set D? false
```

Q2 [1 mark]:

The power set of a set *S* is the set of all subsets of *S*. The power set contains *S* itself and the empty set because these are both subsets of *S*.

For example, the power set of the set {1, 2, 3} is {{1, 2, 3}, {1, 2}, {1, 3}, {2, 3}, {1}, {2}, {3}, Ø}. The power set of a set *S* is usually written as *P*(*S*).

Create a method, named *PowerSet()* in *Set* class to return a new set contains all power sets of *this* set.

Then, test PowerSet() method in Tester class using Set B.

```
Set B: {1, 2, 5, 9, 13}
P(S) of Set B: {{ }, {1}, {2}, {5}, {9}, {13}, {1, 2}, {1, 5}, {2, 5}, {1, 9}, {2, 9},
{5, 9}, {1, 13}, {2, 13}, {5, 13}, {9, 13}, {1, 2, 5}, {1, 2, 9}, {1, 5, 9}, {2, 5, 9},
 {1, 2, 13}, {1, 5, 13}, {2, 5, 13}, {1, 9, 13}, {2, 9, 13}, {5, 9, 13}, {1, 2, 5, 9},
{1, 2, 5, 13}, {1, 2, 9, 13}, {1, 5, 9, 13}, {2, 5, 9, 13}, {1, 2, 5, 9, 13}}
```

Q3 [1 mark]:

Inclusion–exclusion principle

In combinatorics (combinatorial mathematics), the **inclusion–exclusion principle** is a counting technique which generalizes the familiar method of obtaining the number of elements in the union of two finite sets; symbolically expressed as:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

The principle is more clearly seen in the case of three sets, which for the sets *A*, *B* and *C* is given by

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

Generalizing the results of these examples gives the principle of inclusion–exclusion. To find the cardinality of the union of *n* sets:

1. Include the cardinalities of the sets.
2. Exclude the cardinalities of the pairwise intersections.
3. Include the cardinalities of the triple-wise intersections.
4. Exclude the cardinalities of the quadruple-wise intersections.
5. Include the cardinalities of the quintuple-wise intersections.
6. Continue, until the cardinality of the *n*-tuple-wise intersection is included (if *n* is odd) or excluded (*n* even).

Using **inclusion–exclusion principle**, create a method, named *TotalElements(array of n Set)* in Tester class to print out principle formula (refer sample output) and return cardinality of the union of *n* sets in the array.

(**Note**: Algorithm using *union* method created in Q1 is prohibited)

Then, suppose we have a Set S = all natural numbers below 500 and it have subsets of Set W, Set X, Set Y and Set Z. Set W = all odd numbers, Set X = all prime numbers, Set Y = all perfect square numbers and Set Z = all multiples of 7.

Test *TotalElements* method using Set W, Set X, Set Y and Set Z in Tester class.

```
Union of all sets = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 8
3, 85, 87, 89, 91, 93, 95, 97, 99, 101, 103, 105, 107, 109, 111, 113, 115, 117, 119, 121, 123, 125, 127, 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 151, 153, 155, 157, 1
59, 161, 163, 165, 167, 169, 171, 173, 175, 177, 179, 181, 183, 185, 187, 189, 191, 193, 195, 197, 199, 201, 203, 205, 207, 209, 211, 213, 215, 217, 219, 221, 223, 225, 227, 229, 23
1, 233, 235, 237, 239, 241, 243, 245, 247, 249, 251, 253, 255, 257, 259, 261, 263, 265, 267, 269, 271, 273, 275, 277, 279, 281, 283, 285, 287, 289, 291, 293, 295, 297, 299, 301, 303
, 305, 307, 309, 311, 313, 315, 317, 319, 321, 323, 325, 327, 329, 331, 333, 335, 337, 339, 341, 343, 345, 347, 349, 351, 353, 355, 357, 359, 361, 363, 365, 367, 369, 371, 373, 375,
 377, 379, 381, 383, 385, 387, 389, 391, 393, 395, 397, 399, 401, 403, 405, 407, 409, 411, 413, 415, 417, 419, 421, 423, 425, 427, 429, 431, 433, 435, 437, 439, 441, 443, 445, 447,
449, 451, 453, 455, 457, 459, 461, 463, 465, 467, 469, 471, 473, 475, 477, 479, 481, 483, 485, 487, 489, 491, 493, 495, 497, 499} + {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239
, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457,
 461, 463, 467, 479, 487, 491, 499} + {1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484} + {7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77
, 84, 91, 98, 105, 112, 119, 126, 133, 140, 147, 154, 161, 168, 175, 182, 189, 196, 203, 210, 217, 224, 231, 238, 245, 252, 259, 266, 273, 280, 287, 294, 301, 308, 315, 322, 329, 33
6, 343, 350, 357, 364, 371, 378, 385, 392, 399, 406, 413, 420, 427, 434, 441, 448, 455, 462, 469, 476, 483, 490, 497} - {3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59,
 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257
, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467,
 479, 487, 491, 499} - {1, 9, 25, 49, 81, 121, 169, 225, 289, 361, 441} - { } - {7, 21, 35, 49, 63, 77, 91, 105, 119, 133, 147, 161, 175, 189, 203, 217, 231, 245, 259, 273, 287, 301
, 315, 329, 343, 357, 371, 385, 399, 413, 427, 441, 455, 469, 483, 497} - {7} - {49, 196, 441} + { } + {7} + {49, 441} + { } - { }

The total elements between Set W, X, Y and Z:
296 elements
```

Q4 [1 mark]:

A partition of a set *S* is a set of nonempty subsets of *S* such that every element *x* in *S* is in exactly one of these subsets. That is, the subsets are pairwise disjoint (meaning any two sets of the partition contain no element in common), and the union of all the subsets of the partition is *S*.

Some explanations on partitions (if you knew, then skip):

- The set { 1, 2, 3 } has these five partitions (one partition per item):
    - { {1}, {2}, {3} }, sometimes written 1|2|3.
    - { {1, 2}, {3} }, or 12|3.
    - { {1, 3}, {2} }, or 13|2.
    - { {1}, {2, 3} }, or 1|23.
    - { {1, 2, 3} }, or 123 (in contexts where there will be no confusion with the number).
- The following are not partitions of { 1, 2, 3 }:
    - { {}, {1, 3}, {2} } is not a partition (of any set) because one of its elements is the empty set.
    - { {1, 2}, {2, 3} } is not a partition (of any set) because the element 2 is contained in more than one block.
    - { {1}, {2} } is not a partition of {1, 2, 3} because none of its blocks contains 3; however, it is a partition of {1, 2}.

Create a method, named *Partitions()* in *Set* class to return a new Set contains all partitions of *this* set.

Test *Partitions* method using Set A intersect Set B and Set B in Tester class.

```
Partitions of Set A intersect Set B
{{{1, 2, 5}}, {{1}, {2, 5}}, {{2}, {1, 5}}, {{1, 2}, {5}}, {{1}, {2}, {5}}}

Partitions of Set B
{{{1, 2, 5, 9, 13}}, {{1}, {2, 5, 9, 13}}, {{2}, {1, 5, 9, 13}}, {{1, 2}, {5, 9, 13}}, {{5}, {1, 2, 9, 13}}, {{1, 5}, {2, 9, 13}}, {{2, 5}, {1, 9, 13}}, {{1, 2, 5}, {9, 13}}, {{9},
{1, 2, 5, 13}}, {{1, 9}, {2, 5, 13}}, {{2, 9}, {1, 5, 13}}, {{1, 2, 9}, {5, 13}}, {{5, 9}, {1, 2, 13}}, {{1, 5, 9}, {2, 13}}, {{2, 5, 9}, {1, 13}}, {{1, 2, 5, 9}, {13}}, {{1}, {2},
{5, 9, 13}}, {{1}, {5}, {2, 9, 13}}, {{1}, {2, 5}, {9, 13}}, {{2}, {5}, {1, 9, 13}}, {{2}, {1, 5}, {9, 13}}, {{1, 2}, {5}, {9, 13}}, {{1}, {9}, {2, 5, 13}}, {{1}, {2, 9}, {5, 13}},
{{2}, {9}, {1, 5, 13}}, {{2}, {1, 9}, {5, 13}}, {{1, 2}, {9}, {5, 13}}, {{1}, {5, 9}, {2, 13}}, {{1}, {2, 5, 9}, {13}}, {{2}, {5, 9}, {1, 13}}, {{2}, {1, 5, 9}, {13}}, {{1, 2}, {5,
9}, {13}}, {{5}, {9}, {1, 2, 13}}, {{5}, {1, 9}, {2, 13}}, {{1, 5}, {9}, {2, 13}}, {{5}, {2, 9}, {1, 13}}, {{5}, {1, 2, 9}, {13}}, {{1, 5}, {2, 9}, {13}}, {{2, 5}, {9}, {1, 13}}, {{
2, 5}, {1, 9}, {13}}, {{1, 2, 5}, {9}, {13}}, {{1}, {2}, {5}, {9, 13}}, {{1}, {2}, {9}, {5, 13}}, {{1}, {2}, {5, 9}, {13}}, {{1}, {5}, {9}, {2, 13}}, {{1}, {2, 5}, {9}, {13}}, {{2},
 {5}, {9}, {1, 13}}, {{2}, {5}, {1, 9}, {13}}, {{2}, {1, 5}, {9}, {13}}, {{1, 2}, {5}, {9}, {13}}, {{1}, {2}, {5}, {9}, {13}}}
```