Fall 2015 Biostat 615 Homework #3 (Total 30 pts)

Due by Wednesday October 28th, 2015 noon as a compressed file “hw3.tar.gz” containing the required C++ source code files for the following four problems. Please read the instruction carefully before you start to work on your homework.

Instruction

- The additional grading will be executed on every day from October 20th to October 27th. The final grading will be executed exactly at 12:10pm, Wednesday October 28th, 2015.
- Your program might get input from `cin` or `argv`, according to the specific description given each problem.
- Make your algorithm as efficient as you can, as some of the test cases may be computationally challenging and your program will be terminated if it does not finish after running for 10 seconds and you will lose the points for those test cases. However, you should always turn in a program even if it is not very efficient so that you can at least get partial points.
- All your programs (e.g., if it is named as `program.cpp`) will be compiled and tested on the server `scs.itd.umich.edu` using the following command:

```
g++ -O -o program program.cpp
```

  where “program” will be replaced by the corresponding program name for each problem.

- For this homework, you are NOT allowed to use your own header files in your submission.

Problem 1 - Detect Square (10 pts)

Write a C++ program `detectSquare.cpp` which, given coordinates of N points in the 2-D X-Y plane (read from `cin`), detects whether there are four points (among the N points provided) which form the four corners of a square with all edges parallel to the X or Y coordinate axes, and prints “Yes” or “No” correspondingly. The input contains 2N integers, with each pair of adjacent numbers representing the X and Y coordinates of a point. Example runs of valid input arguments are

```
user@host:~/Private/biostat615/hw3$ ./detectSquare
1 2 1 3 2 2 2 3
Yes
user@host:~/Private/biostat615/hw3$ ./detectSquare
1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8
No
```

Problem 2. Exponentiation (10 pts)

Write a C++ program `exponent.cpp` which, given a sequence of nonnegative integers $n_1, n_2, \ldots, n_k$ (read from `cin`), computes and prints the last k digits of m_k, where $m_i = (m_{i-1})^{n_i}$ for $i = 2, \ldots, k$ and $m_1 = n_1$. If the last k digits are all zeros, print "0"; otherwise, omit any leading zeros. Example runs of valid input arguments are

```
user@host:~/Private/biostat615/hw3$ ./exponent
10 5
0
user@host:~/Private/biostat615/hw3$ ./exponent
2 4 4
536
```

Problem 3 - Change Money (10 pts)

Suppose you are traveling in a country where currency is available only in coins of a number of different integer values $0 < n_1 < n_2 < \cdots < n_k$. You need to write a C++ program `changeMoney` (your C++ source code file must be named as `changeMoney.cpp`) which, given an integer number $N > 0$, lists different ways that an amount of money of N can be provided in different combinations of coins. You may assume that all input arguments are positive integer numbers and N is the first argument followed by $n_1, \ldots, n_k$ as the remaining arguments. The first line of your output is the total number of combinations. If the number is positive, starting from the second line, list all the specific combinations, where the coin values are listed in an increasing order within each combination. The order of the combination is determined by coin frequency (list the combinations with more small coins first). Example runs of valid input arguments are

```
user@host:~/Private/biostat615/hw3$ ./changeMoney 5 1 2
3
1 1 1 1 1
1 1 1 2
1 2 2
user@host:~/Private/biostat615/hw3$ ./changeMoney 13 2
0
```

使用 map<int,int>comb ie comb[2]++;来记录面值2的个数
然后再vector<map<int,int>>来输出
然后按照最大面值最小个数优先排序