VE280 Lab2

Out: 12:00, May 25, 2020; **Due**: 23:59, June 01, 2020

Ex1. Eratosthenes Sieve

Problem: At last, Kyon discovered that 775249 is the product of 179, 61 and 71, which are exactly Koizumi's height, weight and waistline. Now Kyon is wondering whether these three numbers are prime numbers. To make things general, given a sequence of integers within a certain range, he wants to find out all the prime numbers without changing their order. Please help him implement a program to complete this task.

Requirement: To check if an integer is a prime, intuitively we can achieve that by dividing integers smaller than itself. However, there is a more efficient algorithm called `Eratosthenes Sieve`, and you are required to implement it for this exercise. The algorithm works as follows:

1. For all integers within given range, mark them as prime
2. For 0 and 1, mark them as not prime
3. Find the next smallest integer `i` that is marked as prime
4. For all integers that can be divided by `i`, mark them as not prime
5. Repeat Steps 3-4 until no more `i` can be found in Step 3

After running the above algorithm, you have generated a lookup table. To check if a given integer is a prime, simply check that lookup table and see if it is marked as prime or not.

Input Format: The first line: an integer n `(1 <= n <= 20)` that stands for the length of the sequence. The second line: n integers within range `[0,100000]`.

```
# sample
3
179 61 71
```

Output Format: All the primes in original order.

```
# sample
179 61 71
```

Tags: `#array` `#time-complexity`

Ex2. Which Building Will Be Destroyed?

Problem: Unfortunately, Kyon's action of Hiding Photos was discovered by Haruhi. Haruhi got so angry that she unconsciously dragged Kyon into a closed space where a celestial was destroying the buildings. The celestial's action could be predicted as follow:

1. Equally divide the square into 4 smaller squares, and the left-upper part will all be destroyed.
2. For each of the remaining 3 smaller squares, also divide them into 4 smaller squares, and the left-upper part will be destroyed
3. Repeat until the square can no longer be divided

Suppose you are Kyon, you need to write a program to calculate which buildings will be destroyed in order to save your own life.

Requirement: You will use a simple recursion to solve this exercise. See `Takeaway` for some hints.

Input Format: An integer n `(1 <= n <= 10)` that stands for a `2^n by 2^n` square.

```
# sample
3
```

Output Format: A `2^n by 2^n` square , where 0 indicates destroyed, 1 indicates safe.

```
# sample
0 0 0 0 0 0 0 1
0 0 0 0 0 0 1 1
0 0 0 0 0 1 0 1
0 0 0 0 1 1 1 1
0 0 0 1 0 0 0 1
0 0 1 1 0 0 1 1
0 1 0 1 0 1 0 1
1 1 1 1 1 1 1 1
```

Tags: `#abstraction` `#specification` `#recursion`

Ex3. Strength Ranking

Problem: Finally, Kyon successfully escaped from the closed space by kissing Haruhi. The huge celestial really reminded Kyon of the big giants in *Attack on Titan*, so he invited Haruhi to watch this anime together. However, they had a intense debate about whether Mikasa or Levi was stronger. They finally decided to rank the soldiers in the Survey Group by the sum of their combat, initiative and wits ability grades. If two soldiers have the same sum grades, just keep the order the same as in the input. Please write a program for them to help them with the rank.

Requirement: You are required to use a `struct` to store information for each soldier. Also, you are required to use `qsort()` in `<cstdlib>` to achieve sorting. Check http://www.cplusplus.com/reference/cstdlib/qsort/ on how to use that.

Input Format: The first line: an integer n `(1<= n <= 10)` that stands for the number of soldiers. The following n lines: Name (string), Combat (int), Initiative (int), Wits (int). You can assume that there is no space in name and the length is less than `16`.

```
# sample
3
Eren 9 10 3
Mikasa 10 9 8
Levi 11 10 8
```

Output Format: Soldiers sorted in descending total score.

```
# sample
Levi 11 10 8
Mikasa 10 9 8
Eren 9 10 3
```

Tags: `#function` `#struct` `#qsort` `#pointer`

Files

2. *lab2_description.pdf*: description in pdf
3. *lab2_starter_files.zip*: starter files

Submission

Compress each **.cpp** file into a **.tar** file and submit to JOJ. Do not change the name of the **.cpp** file.

For example, when submitting ex1, your file name should be ex1.cpp. You should compress ex1.cpp into ex1.tar and submit ex1.tar to JOJ.