

Problem J4/S1: Party Invitation

You are hosting a party and do not have room to invite all of your friends. You use the following unemotional mathematical method to determine which friends to invite.

Number your friends $1, 2, \dots, K$ and place them in a list in this order. Then perform m rounds. In each round, use a number to determine which friends to remove from the ordered list.

The rounds will use numbers r_1, r_2, \dots, r_m . In round i remove all the remaining people in positions that are multiples of r_i (that is, $r_i, 2r_i, 3r_i, \dots$) The beginning of the list is position 1.

Output the numbers of the friends that remain after this removal process.

Input

The first line of input contains the integer K ($1 \leq K \leq 100$). The second line of input contains the integer m ($1 \leq m \leq 10$), which is the number of rounds of removal. The next m lines each contain one integer. The i -th of these lines ($1 \leq i \leq m$) contains r_i ($2 \leq r_i \leq 100$) indicating that every person at a position which is multiple of r_i should be removed.

Output

The output is the integers assigned to friends who were not removed. One integer is printed per line in increasing sorted order.

Party Invitation Pseudo Code

```
cin >> size;

list[size];

//initiate array (1 – size)

int currentSize = size;

while (round > 0){

    cin >> removeIndex;

    for ( i = removeIndex – 1; i < size; i += removeIndex){

        //erase element at i – numRemoved;

        //currentSize --;

        //numRemoved ++;

    }

    size = currentSize;

}
```

Problem J3: Clock Hands

Alice had just dug up an old time capsule that she had buried in her backyard many years ago. Inside, she discovered an exquisite clock from when she was a young girl. When she buried the clock, she had removed the batteries so that the hands would remain in that position forever. This was done so she could remind her future self of the exact time at which she met her eternal love interest — Bob. Unfortunately, all but the 12 o'clock marking had worn away over the years.

With a protractor handy, Alice is determined to find out the exact time that Bob came into her life.

Measuring the hour hand from the centre of the clock, Alice determined that it was H degrees clockwise from the 12 o'clock marking. Alice then measured the angle of the minute hand, determining that it was M degrees clockwise *relative to the hour hand*. Finally, she measured the second hand, determining that it was S degrees clockwise *relative to the minute hand*. The three angle measurements will all be non-negative real numbers greater than or equal to 0, but less than 360. You may assume that the angles will yield valid times.

Input Format

Three lines, containing the three values H , M , and S .

Output Format

Output the time, in the format HH:MM:SS, where HH ($1 \leq HH \leq 12$) is the number of hours, MM ($0 \leq MM < 60$) is the number of minutes, and SS ($0 \leq SS < 60$) is the number of seconds. These values each must be padded to two-digits with leading zeros. Round the time to the nearest second.

Sample Input 1

195 345 180

Sample Output 1

06:30:00

Sample Input 2

90.066667 270.733333 47.200000

Clock Hands Pseudo Code

```
cin >> angleHour >> angleMin >> angleSecond;
```

```
//convert angle relatively to 12 o'clock under 360 degree
```

```
angleHour = angleHour - (int)(angleHour / 360) * 360; //angle - (numOf360 in angle) * 360
```

```
....
```

```
//convert angle to time
```

```
hour = 12/360 * angleHour; //12hr / 360 degree * angleHour
```

Problem J2: Rope Unfolding

Alice's love potion failed because Bob overdosed. Now, her new tactic is to impress Bob. Bob is a computer scientist who specializes in ropes. In computer science, a rope is a data structure used for efficiently manipulating strings. Bob has a rope of characters. However, it's too long to fit in his pocket, so he folds it into N segments. Each fold is at a letter which is not the last. The fold takes the rest of the rope after that letter, reverses it, and puts it above the current letter. For example, the rope "ABRACADABRA", when folded "AB|RACA|DAB|RA", yields:

AR DAB ACAR AB

Alice notices that Bob often gets a headache pulling the rope out of his pocket and spending a long time untangling it. She concludes that if she finds a way to unfold it for Bob, Bob will be so impressed that he can't help but fall madly in love with her. You must write a program to help Alice!

Input Format

The first line of input will contain the positive integer N ($1 \leq N \leq 100$), the number of lines to follow. The next N lines each contain a segment of the folded rope.

The entire, unfolded rope will be no greater than 100 characters, and will only consist of uppercase letters from 'A' to 'Z'.

Output Format

The output should contain one line — the unfolded rope.

Rope Unfolding Pseudo Code

```
if (even) left = false; //start from right
```

```
if (odd) left = true; //start from left
```

```
while(numLinesLeft > 0){
```

```
    //reverseCurrentLine
```

```
    //if(!left) attach reversed line to the right
```

```
    //if(left) attach reversed line to the left
```

```
    left = !left;
```

```
}
```

Problem J4: Selecting Shifts

Alice figured out where Bob works, and has obtained a job there so she can secretly admire him at work. Because she is new, the position is only part-time, and she is always called in to shifts by her boss. Whenever this happens, she is offered various shifts to pick from, starting and ending at different times of the day. Alice knows that Bob works from $T1$ to $T2$ on days that she's called in.

Alice's employer has given her a list of the shifts that she can choose from in order of decreasing pay. Given this list, Alice would like to know which one would result in the longest period of time that both of them are simultaneously on the job. If multiple shifts yield the same time spent with Bob, she would like to know the highest paying out of those (i.e. the one that appears earliest in the list).

Input Format

Line 1 contains $T1$ and $T2$, the time that Bob starts and ends his shift.

Line 2 contains the integer N ($1 \leq N \leq 100$), the number of shifts that Alice is being offered.

The next N lines each contain two times — the start and end times of that shift. The shifts are ordered by decreasing pay.

Times in the input will be in the 7 character long format as depicted in the sample input below, and will all describe times within the same 24 hour day (between 12:00AM and 11:59PM, inclusive). Every shift described will have the start time strictly before the end time.

Output Format

The start and end time of the shift that Alice should pick to maximize the time she gets to spend on the job with Bob. If there are multiple answers, pick the one that is higher up on the list. If none of the shifts overlap with Bob's shift, output "Call in a sick day." (without quotes), since Alice has no interest in wasting her time at work when Bob is not there.

Sample Input

```
10:30AM 07:00PM 4 05:30AM 10:00AM 11:00AM 08:15PM 01:45PM 05:00PM 08:00PM 11:00PM
```

Sample Output

```
11:00AM 08:15PM
```

Shift

```
int numConverted = atoi(char[])
```

- if we have a string rather than an array of char, then we can convert string to char[] by using:
yourString.c_str() ➔ int num =
atoi(yourString.c_str());

SpaceTime (Optional)

- 3D Array Traversal
- Use breath first search (This will be covered in the last lecture). This is widely used to find the shortest path. And with the right heuristics, we can develop different algorithm to find path.
- Will come back to this question in the last lecture.

Vectors

In java called List:

<http://docs.oracle.com/javase/7/docs/api/java/util/List.html>

Vector

- Example motivation:
 - Want to read integers from user input
 `> 1 -20 3 31 55 <Ctrl+D>`
 - Then pass the array on to the rest of the program to process, print, examine, ...
 - **Problem**: how big an array should we allocate?
 `int *array = new int[??];`
 - Don't know until **after** the input is read!

Vector

- Wouldn't it be great to have something **just like an array** that **could grow** as needed?

```
#include <vector>
#include <iostream>
using namespace std;

vector<int> get_input ( ) {
    vector<int> vec;           // vector of ints, initially empty
    int val;
    cin >> val;
    while ( val != 0 ) {
        vec.push_back (val);  // Add new value to end of vector
        cin >> val;
    }
    return (vec);              // Return the whole vector
}
```

How you declare vector

main.cpp

Declaration

```
#include <vector>
using namespace std;

int main () {
    vector<int> vec1; //size = 0
    vector<int> vec2(10); // size = 10, but values unknown
    vector<int> vec3(10,-1); // size = 10, all values -1
}
```

Vector: Can Use Like Array

main.cpp

```
#include <vector>
#include <iostream>
using namespace std;
```


```
. . .
```

```
int main () {
    vector<int> in_vals;
    in_vals = get_input ();
    for (int i = 0; i < in_vals.size(); i++)
        cout << in_vals[i] << " ";
```

```
    cout << endl;
```

```
    for (int i = in_vals.size() - 1; i >= 0; i--)
        cout << in_vals[i] << " ";
}
```

vectors know how many elements they contain. Valid data from index 0 to size()-1



How would I print out the vector in reverse order?

Input: 1 -20 3 31 55 <Ctrl+D>

Output: 1 -20 3 31 55
55 31 3 -20 1

Vector Functions

`grades[5]` ← Accessing the first element

`cout << grades.size() << endl;` ← Prints the size of the vector

`values.push_back(1);` ← Adds the integer 1 to the back of values

`values.pop_back();` ← Removes the last element from values

There is no `pop_front`.

Rest is available at

<http://www.cplusplus.com/reference/vector/vector/?kw=vector>

Practice/Homework

- <https://dmoj.ca/problem/ccc07s2>
- <https://dmoj.ca/problem/ccc03s1>
- <https://dmoj.ca/problem/ccc05s1>
- <http://wcipeg.com/problem/ccc05s3>