Introduction

In this problem, you will be cracking RSA. This assignment will require you to write a program in C++ to the specifications below, and will be graded along with your LaTeX submission of Homework 2. In order to get full credit, you will need to pass all test cases. The first 3 are provided below, and the remaining ones will be hidden.

Submission Information

- You will submit a single .cpp file on Gradescope under "RSA Code"
- There are 16 test cases in total, 3 visible and 13 hidden, and all must be passed for full credit
- You can submit as many times as you wish
- You may use external resources pertaining to C++ and its syntax, standard libraries, etc.
- You may use C++ standard libraries
- You may NOT use external resources for any purpose outside of those listed above, and you may not use code from external resources, even with citation. If you have a question about what is permitted, ask a TA or the professor
- You may ask a TA (Ezekiel) for help, but you will not be provided the input for hidden test cases

Tips

- Make sure to check every way a key can be considered invalid; these are outlined in the RSA specification
- Make sure all integers are correctly mapped to characters
- Make sure that your output matches the autograder output exactly, including whitespace, letter case, and any user prompts you may have added
- Keep integer overflow in mind
- If the autograder times out, your code has an infinite loop
- Make sure you're explicitly including libraries for any functions used, as the compiler will not add it for you
- Use vectors to store the message at each stage (raw input, decrypted integers, and text)
- Efficiency doesn't matter too much, so brute force may be a good option for some parts

Crack the Code

Alice and Bob like sending each other secret quotes. Today is Bob's turn and he's planning on sending Alice a quote from a famous person.
Bob also learned about the RSA encryption scheme in his CS111 class and he decided to try it out. He first chooses a number to start with x and converts his quote by assigning numbers to the letters as such:

- $A = x, B = x + 1, ..., Z = x + 25$
- Blanks $= x + 26$
- Quotation marks $= x + 27$

- Comma $= x + 28$
- Period $= x + 29$
- Apostrophe $= x + 30$

This time, he chooses x to be 7. Alice then sends him the public key $P = (e, n)$ so that he can encrypt his message, and Bob sends the encrypted message to Alice. However, you are able to intercept both the public key and the encrypted message.
Can you crack the code and decrypt Bob's message?

Input

The first line will have two values, e, n which represent the public key.
The second line will have one value m which is the length of the encoded message.
The next m integers will represent the encoded message.
Take all the input in using standard input (for C++ it's `cin`).

Output

If the public key is invalid, print out the error message shown in the example below.
Otherwise:
In the first line, output p, q, $\phi(n)$, and d. Note that it should be that $p < q$, $p \cdot q = n$ and $\phi(n) = (p-1)(q-1)$.
In the second line, output the decoded message in integers with a space in between.
In the third line, output the decoded message in characters. If the characters are in the English alphabet, make sure to print them out uppercase.
Output your solution using standard output (for C++ it's `cout`).

Examples

Test Case 1. Input:

```
23 55
13
34 13 17 14 10 34 22 53 11 23 52 49 53
```

Output:

```
5 11 40 7
34 7 8 9 10 34 33 37 11 12 13 14 37
"ABCD" 'EFGH'
```

For the input, the public key is $P = (e, n) = (23, 55)$ and the encrypted message is of length 13. From the output, we get $p = 5$, $q = 11$, $\phi(n) = 40$, and $d = 7$. We also obtain the decrypted message, which we convert back to text in the following line.

Test Case 2. Input:

```
5 55
13
34 32 43 34 10 34 33 12 11 12 43 34 12
```

Output:

```
Public key is not valid!
```

For the input, the public key is $P = (e, n) = (5, 55)$ and the encrypted message is of length 13. This time the program outputs the error message because the inputted public key was invalid. In this case it is because $d = e^{-1} \pmod{\phi(n)}$ does not exist.

Test Case 3. Input:

```
2397 5353
58
2552 998 4327 4012 1967 4394 4327 4986 3743 3667 4012 3646 3646 4012 3646 4327 3558 3743
4012 4012 4986 4012 4327 4986 4394 3667 1967 998 2731 3743 4394 4327 3177 3667 2942 3490
4327 4394 3743 4012 4327 332 1967 2731 4327 1050 3743 4012 178 4327 998 2997 3490 4327
4986 1967 3646 2552
```

Output:

```
53 101 5200 2933
34 15 33 11 7 26 33 25 14 24 11 10 10 11 10 33 9 14 11 11 25 11 33 25 26 24 7 15 13 14 26 33
12 24 21 19 33 26 14 11 33 8 7 13 33 29 14 11 20 33 15 37 19 33 25 7 10 34
"I EAT SHREDDED CHEESE STRAIGHT FROM THE BAG WHEN I'M SAD"
```

For the input, the public key is $P = (e, n) = (2397, 5353)$ and the encrypted message is of length 58. From the output, we get $p = 53$, $q = 101$, $\phi(n) = 5200$, and $d = 2933$. We also obtain the decrypted message, which we convert back to text in the following line.