

소수로 만든 숫자

Description

N개의 숫자를 입력으로 받아 이진수 B를 만들고자 한다.

첫 번째 숫자가 소수이면 B의 첫 번째 비트(MSB, most significant bit)가 1이고, 아니면 0이다.

N 번째 숫자가 소수이면 B의 마지막 비트(LSB, least significant bit)가 1이고, 아니면 0이다.

i 번째 숫자가 소수이면 B의 i 번째 비트가 1이고, 아니면 0이다.

예를 들어, N = 5 이고, 주어진 숫자가 1 2 3 4 5 라면 01101 = 13(십진수) 이다.

N개의 숫자를 입력으로 받아, 소수로 만든 숫자 B를 16진수로 출력하시오.

```
#include <stdio.h>
#include <math.h>

int is_prime(unsigned int n);
unsigned int solve(unsigned int N);

int main()
{
    unsigned int N;
    scanf("%d", &N);
    printf("%X\n", solve(N));
}
```

Input

첫 번째 줄에 양의 정수 N이 주어진다. (1 <= N <= 32)

두 번째 줄에 N개의 양의 정수 M이 주어진다. (1 <= M <=2^32 - 1)

Output

첫 번째 줄에 양의 정수 B를 16진수로 출력한다.

16진수는 대문자로 표기한다. (%X 포맷 이용)

Sample Input 1

5
1 2 3 4 5

Sample Output 1

D

Sample Input 2

32
32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 1
0 9 8 7 6 5 4 3 2 1

Sample Output 2

50451456

```
1 int is_prime(unsigned int n)
2 {
3     if (n <= 1)
4         return 0;
5     if (n <= 3)
6         return 1;
7
8     if (n % 2 == 0)
9         return 0;
10
11     for (unsigned i = 3; i <= (unsigned int)(sqrt(n)); i += 2)
12     {
13         if (n % i == 0)
14         {
15             return 0;
16         }
17     }
18     return 1;
19 }
20
21 unsigned int solve(unsigned int N)
22 {
23     int k = 0;
24
25     for (unsigned int i = 0; i < N; i++)
26     {
27         unsigned long long temp;
28         scanf("%llu", &temp);
29         if (is_prime(temp))
30         {
31             k |= (1 << (N - 1 - i));
32         }
33     }
34     return k;
35     // 숫자 다 넣음.
36 }
```