

Assignment 13: A Bit Set

13.1 Bitwise Operators

Using the bitwise XOR operator, you may swap the values of two integer variables without any temporary variable. The following code shows an example code of how to do it:

```
1 #include <stdio.h>
2
3 void swap(int *a, int *b)
4 {
5     *a ^= *b, *b ^= *a, *a ^= *b;
6 }
7
8 int main()
9 {
10     int n = 123;
11     int a = n >> 1, b = n << 1;
12
13     printf("a:%3X\tb:%3X\n", a, b);
14     swap(&a, &b);
15     printf("a:%3X\tb:%3X\n", a, b);
16
17     return 0;
18 }
```

To prepare the values of two variables, the bitwise shift operators are used. Note that the shift operations in Line 11 correspond to dividing by two and multiplying by two, respectively.

13.2 Programming Assignment 13: bitset.c

Considering the binary representation, a positive integer can be considered as a set. For example, 2 can represent a set $\{1\}$ and 5 can represent a set $\{0\}$. Can you guess what? That's right! The set corresponds to the indexes (starting from 0) of bits of the binary representation of the number. For the example cases, the following equations hold:

$$\begin{aligned} 2 &= 2^1 \\ 5 &= 2^2 + 2^0 \end{aligned}$$

Let's call this set as the *bitset* of the integer. The bitset of n is denoted by $B(n)$.

Write a program reading a positive integer n and printing the elements of $B(n)$ line by line in the increasing order. Calculating $B(n)$, your program should use the bitwise operators rather than arithmetic operators.

The program is to read from standard input. The input consists of a single line containing an integer n ($n > 0$). The program should print to standard output. The output consists of several lines, where each line contains an element of $B(n)$. The order of the data items should be in increasing order.

Additional requirements for bonus points

- Do not use any loops in your program.
- Do not use `if` or `switch` statement.

Input	Output
2	1
5	0 2
123	0 1 3 4 5 6