

## 2.13小测试

测试时间 2022年2月13日 08: 00 ~ 12: 00

题目	防签到题	翻转游戏	字符清除	书本分配
文件名	anti.cpp	game.cpp	string.cpp	book.cpp
输入输出	anti.in/.out	game.in/.out	string.in/.out	book.in/.out
时间限制	1s	1s	1.5s	2s
空间限制	128MB	128MB	256MB	256MB
编译命令	-O2 -std=c++14	-O2 -std=c++14	-O2 -std=c++14	-O2 -std=c++14

# T1: 防签到题(anti)

文件名 anti.cpp/.in/.out ，时间限制1秒，空间限制128MB，编译开启 -O2 优化，支持 C++14。

## 问题描述

系统中有一个神奇函数：

```
unsigned func(unsigned n) {  
    unsigned x = n & -n;  
    return (n + x) | ((n ^ (n + x)) / x) >> 2;  
}
```

但由于数据类型的限制，这个函数只适用于 32 位无符号整数的计算。请你实现同样功能的支持  $k$  位无符号整数的函数。

## 输入格式

本题有多组测试数据，第一行输入数据组数  $T$ 。

每组数据输入 2 个正整数  $n, k$ 。

## 输出格式

每组数据输出一行一个整数，表示计算结果。

## 样例1

样例输入

```
3  
1 32  
10 32  
100 32
```

样例输出

```
2  
12  
104
```

## 数据范围

对于 30% 的数据,  $k = 16$

对于 60% 的数据,  $2 \leq k \leq 32$

对于 100% 的数据,  $2 \leq k \leq 64, 0 < n < 2^k, 1 \leq T \leq 1000$

## T2: 翻转游戏(game)

文件名 `game.cpp/.in/.out` , 时间限制1秒, 空间限制128MB, 编译开启 `-O2` 优化, 支持 C++14。  
本题为 *spj* 评测。

### 问题描述

一个长为  $n$  的 01 串, 其中前  $k$  个字符是 1, 其余为 0。

你要进行  $m$  次 (需满足  $0 \leq 2^{m-1} \leq n$ ) 下列操作, 最终将该串变为全 0:

- 在第  $i$  次操作中, 你 **必须** 选择一个长度为  $2^{i-1}$  的子串, 将其翻转, 即 0 变为 1、1 变为 0

### 输入格式

本题有多组测试数据, 第一行输入数据组数  $T$ , 对于每组数据:

输入两个整数  $n$ 、 $k$ 。

### 输出格式

对每组数据, 若无解, 输出 NO; 有解输出 YES, 并输出任意一种方案, 格式如下:

首先输出一个非负整数  $m$ , 表示操作的次数 (要保证  $2^m \leq n$ )。

接下来一行输出  $m$  个正整数  $p_1, p_2, \dots, p_m$ , 表示每次操作中选择的子串的左端点, 即第  $i$  次选择翻转的区间为  $[p_i, p_i + 2^{i-1} - 1]$ , 因此要保证  $1 \leq p_i \leq n - 2^{i-1} + 1$ 。

### 样例1

样例输入

```
3
5 0
3 3
2 2
```

样例输出

YES  
0  
YES  
2  
3 1  
NO

样例解释

第二组数据中，翻转过程如下：

111  
第一次操作，翻转[3，3]  
110  
第二次操作，翻转[1，2]  
000

数据范围

对于 10% 的数据： $n \leq 20$

对于 60% 的数据： $1 \leq n \leq 10^9, 0 \leq k \leq n/2$

对于 100% 的数据： $1 \leq T \leq 5000, 1 \leq n \leq 10^9, 0 \leq k \leq n$

## T3: 字符清除(string)

文件名 `string.cpp/.in/.out` , 时间限制1.5秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

### 问题描述

何老板给你一个长度为  $n$  的字符串  $S$  , 他将进行  $m$  次字符清除操作。

每次操作, 他会给出三个参数 " $L\ R\ C$ ", 表示把当前  $S$  下标区间为  $[L, R]$  的字符  $C$  都全部清除掉, 每次操作执行完后,  $S$  中剩余的字符会自动合拢。

$m$  次操作后, 请你输出字符串  $S$  。

### 输入格式

第一行, 两个整数  $n, m$ 。

第二行, 字符串  $S$  (由小写字母、大小字母和数字构成)

接下来  $m$  行, 每行三个参数 " $L\ R\ C$ " 表示一次清除操作(字符串下标从 1 开始)

注意: 设当前字符串长度为  $n'$ , 如果  $L > n'$ , 忽略本次操作。如果  $R > n'$ ,  $R$  只操作到  $n'$  位置。

### 输出格式

一行, 最后得到的字符串  $S$  。

### 样例1

样例输入

```
4 2
abac
1 3 a
2 2 c
```

样例输出

```
b
```

### 样例2

样例输入

```
9 5
aAAaBBccD
1 4 a
5 6 c
2 3 B
4 4 D
2 3 A
```

样例输出

```
AB
```

### 样例3

样例输入

```
10 4
agtFrgF4aF
2 5 g
4 9 F
1 5 4
1 7 a
```

样例输出

```
tFrg4
```

### 数据范围

- 对于 10% 的数据： $1 \leq n, m \leq 1000$
- 对于 30% 的数据： $1 \leq n, m \leq 10000$
- 对于 100% 的数据： $1 \leq n, m \leq 200000$

# T4: 书本分配(book)

文件名 `book.cpp/.in/.out` , 时间限制2秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

## 问题描述

$n$  本书分配给班上的  $m$  位同学, 从编号为 1 的书开始连续分配, 每人至少分到一本书, 且分到的书的编号必须是连续的。

可以剩余一些书不进行分配。

每本书有阅读后的快乐值  $a_i$ , 每位学生的快乐值为所分配的书本快乐值之和。

合理分配书本, 使得学生获得的快乐值中的最大值最小。

## 输入格式

第一行输入 2 个正整数  $n, m$ 。

接下来 1 行输入  $n$  个整数  $a_i$ 。

## 输出格式

输出在合理分配方案下学生获得的快乐值中最大值的最小值。

## 样例1

样例输入

```
4 2
3 -2 4 -2
```

样例输出

```
2
```

## 数据范围

对于 30% 的数据,  $1 \leq m \leq n \leq 10^3$

对于 50% 的数据,  $1 \leq m \leq n \leq 10^4$

对于 80% 的数据,  $1 \leq m \leq n \leq 10^5$



对于 100% 的数据,  $1 \leq m \leq n \leq 2 \times 10^5, -10^9 \leq a_i \leq 10^9$