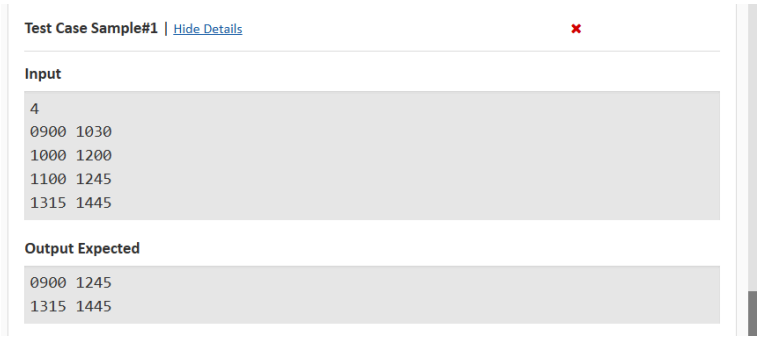


Conference Scheduler

You are working as a scheduling coordinator at a conference center. Your task is to optimize room usage by merging overlapping time slots to avoid booking conflicts.

Input:

- First line contains n, the number of booking intervals
- Next n lines contain two space-separated integers representing start_time and end_time
- Times are represented in 24-hour format (e.g., 1300 for 1:00 PM)

Output:

- Merged list of non-overlapping intervals in sorted order
- Each interval displayed in a readable time format

Constraints:

- 0000 ≤ start_time < end_time ≤ 2400
- start_time and end_time follow valid 24-hour time format

Sample Input	Sample Output
4	0900 1245

Samraj And Bits

Samraj loves to play with numbers. Somebody gave him some numbers and he wishes to play a game with them. There are 3 operations that have to be applied periodically on the given set of numbers.

The operations to be applied in periodic manner are (bitwise) XOR, ADD, (bitwise)OR.

Suppose there are five numbers.

Operations will be applied in the following manner: **((((a XOR b) ADD c) OR d) XOR e)**, and it will go on further for more numbers in a periodic fashion.

You have to predict whether the maximum value possible by applying those operations cyclically from any chosen subset of numbers from the given set is odd or even.

NOTE : **The sequencing of the numbers in the subset will be preserved from the main set**.

Input Format:

The first line consists of an integer T.

T testcases follow. The first line of each testcase consists of an integer N.

In next N lines, each line will contain 1 integer denoting the elements of the array.

Constraints:

1 <= T <= 10

1 <= N <= 16

0 <= A[i] <= 10^5

Output Format:

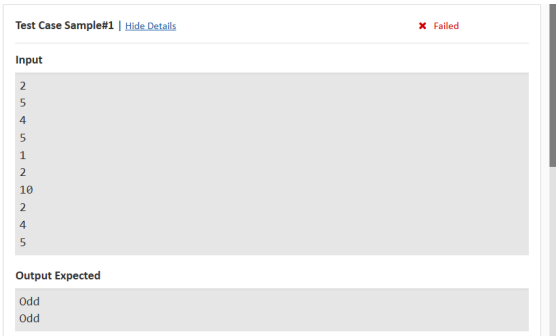
For each testcase, print the answer in a new line i.e. either "**Odd**" (without the quotes) or "**Even**" (without the quotes).

Sample Input

```
2
5
4
5
```

Sample Output

```
Odd
Odd
```



Test Case Sample#1 | Hide Details ✖ Failed

Input

```
2
5
4
5
1
2
10
2
4
5
```

Output Expected

```
Odd
Odd
```

Test Case Sample#1 | Hide Details ✖ Failed

Input

```
2
3
1 2 3
1
9
```

Output Expected

```
-4
9
```

Output of your code

```
2
3
1 2 3
1
9
```

Matt and the gift

Matt likes cooking. But more than that, he likes to give gifts. And now he wants to give his girlfriend an unforgettable gift. But unfortunately he forgot the password to the safe where the money he saved for the gift is kept.

But he knows how to hack the safe. To do this, you need to correctly answer questions asked by the embedded computer. The computer is very strange, and asks special questions, sometimes it can ask about 10000 question (really weird). Because of this, Matt wants you to write a program that will help him to crack the safe.

The questions are different, but there is only one type of question. Several numbers are given and between them one of three characters: *, +, - can be inserted. Note that in this case there is no priority for the operators, that is, if + is the before multiplication, you must first execute the operation of addition, and then multiplication (1 - 2 * 3 must be interpreted as (1 - 2) * 3 = -3 and not -5). The computer asks the minimum possible value of any valid expression.

Input:

The first line of the input contains an integer T denoting the number of test cases. The first line of each test case contains a positive integer N. The second line contains N space separated integers A1, A2, ..., AN denoting the expression without the operators.

Output:

For each test case, output a single line containing the minimal value of given expression.

Constraints:

$1 \le T \le 10^5$

$1 \le N \le 10$

$-9 \le A[i] \le 9$

Sample Input

2

3

1 2 3

1

9

Sample Output

-4

9

Explanation

Example case 1: 1-2-3 = -4

Sample Output

Sample Input

Sample Output