1. Mr. Omuk thinks that Lucky Numbers are numbers where every digit of that number is either even or odd. Mr. Omuk needs a program to detect whether a number is a lucky number or not according to the definition given by him. Now, he wants you to write a program for him. For better understanding, Mr. Omuk has shared a sample input-output with you. Now, write a program in C programming language that takes an integer as input and checks whether it is a lucky number or not.

Input	Output
55555	Lucky Number
2469	Not a Lucky Number
121	Not a Lucky Number
262	Lucky Number

2. Robin Hood likes to loot rich people since he helps the poor people with this money. Instead of keeping all the money together he does another trick. He keeps n sacks where he keeps this money. The sacks are numbered from **0 to n-1**.

   Now each time he can he can do one of the three following tasks:

   I. Give all the money of the i th sack to the poor, leaving the sack empty (**i^{th}** element will be 0).
   II. Add new amount (given in input) in the ith sack.
   III. Find the total amount of money from **i^{th}** sack to **j^{th}** sack.

   Since he is not a programmer, he seeks your help.

   Input

   I. First Line of Input contains two integers n and q.
   II. The next line will take n space separated integers in the range. Here, the **i^{th}** integer denotes the initial amount of money in the **i^{th}** sack.
   III.

IV. Then it will take q lines of more input. Each of the lines contains a task in one of the following form:

- ***1 i*** - give all the money of the $\mathbf{i^{th}}$.
- ***2 i v*** - add money v ($1 \le v \le 1000$) to the $\mathbf{i^{th}}$ ($0 \le i < n$) sack.
- ***3 i j*** - find the total amount of money from the $\mathbf{i^{th}}$sack to the $\mathbf{j^{th}}$sack ($0 \le i \le j < n$).

Output

If the query type is 1, then print the amount of money given to the poor. If the query type is 3, print the total amount from $\mathbf{i^{th}}$ to $\mathbf{j^{th}}$ sack. After all the query print the current condition of the sacks.

Input	Output	Explanation
5 6 3 2 1 4 5 1 4 2 3 4 3 0 3 1 2 3 0 4 1 1	5 14 1 13 2 3 0 0 8 0	3 2 1 4 5 1st Query : 3 2 1 4 0 [Output 5] 2nd Query:3 2 1 8 0(add 4 with 4)[No Output] 3rd Query : 3 2 1 8 0 [Output : 3+2+1+8=14] 4th Query : 3 2 0 8 0 [Output 1] 5th Query : 3 2 0 8 0 [Output : 3+2+0+8+0=13] 6th Query : 3 0 0 8 0 [Output 2] [Output 3 0 0 8 0]

3. Most programmers will tell you that one of the ways to improve your performance in programming is to Practice a lot of Problems. Farina took the above advice very seriously and decided to set a target for herself. Farina decides to solve **at least 10** problems every week for **44 weeks**. She always keeps her problems in a folder named **W1 … W44**. She needs to develop a program which will help her to know about her completions rate towards her goal.

Input Format

The First Line takes **T** which is the number of test cases (the number of times the whole operation of the following lines will execute) for Farina. Each of **T** times the following lines happen:

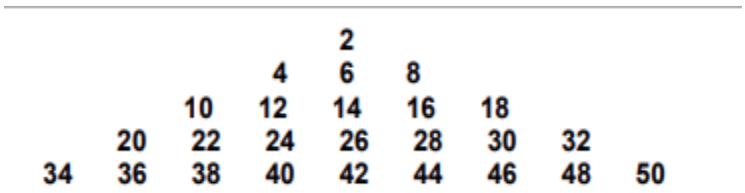
- First Line takes the number of weeks **W** completed so far by Farina.
- Next line takes **W** space separated integers indexed **0 to W-1** which denotes the number of problems solved in $\mathbf{i^{th}}$ week.
- Next Line takes the range of Weeks **R1** and **R2** She wants to Search.

Output Format

For Each Test case there will be a new line of output. In Each line, the index of the weeks which satisfies Farina's target will be displayed as space separated integers.

Input	Output	Explanation
2 5 10 11 2 44 3 2 4 6 1 1 1 23 44 2 1 5	3 3 4	**Input: Line 1:**The whole operation will operate 2 times **(For operation 1)Line 2:** 5 weeks have been completed **Line 3:**The number of solved problems of 5 weeks **Line 4:** 2 to 4 weeks is searched **Output: Line 1:** the number of solved problem of 2 nd index=2, 3 rd index =44, 4th index=3; so only 3 rd index satisfies the condition so it will print only 3. Similarly the next operations will execute.

4. Write a C program to print the following pattern.

Input	Output
5	
3