

1. Ampersands

Program Name: Ampersands.java **Input File:** none

Input File: none

James wants to print a design of ampersands (&) to use as a pattern for an art project. You are to print his design as shown below.

Input

There is no input for this problem.

Output

You are to output a triangle of ampersands (&) with 20 rows as shown below.

Output to Screen

𐤀 𐤁 𐤂 𐤃 𐤄 𐤅 𐤆 𐤇 𐤈 𐤉 𐤊 𐤋 𐤌 𐤍 𐤎 𐤏 𐤐 𐤑 𐤒 𐤓 𐤔 𐤕 𐤖 𐤗 𐤘 𐤙 𐤚 𐤛 𐤜 𐤝 𐤞 𐤟 𐤠 𐤡 𐤢 𐤣 𐤤 𐤥 𐤦 𐤧 𐤨 𐤩 𐤪 𐤫 𐤬 𐤭 𐤮 𐤯 𐤰 𐤱 𐤲 𐤳 𐤴 𐤵 𐤶 𐤷 𐤸 𐤹 𐤺 𐤻 𐤼 𐤽 𐤾 𐤿 𐥀 𐥁 𐥂 𐥃 𐥄 𐥅 𐥆 𐥇 𐥈 𐥉 𐥊 𐥋 𐥌 𐥍 𐥎 𐥏 𐥐 𐥑 𐥒 𐥓 𐥔 𐥕 𐥖 𐥗 𐥘 𐥙 𐥚 𐥛 𐥜 𐥝 𐥞 𐥟 𐥠 𐥡 𐥢 𐥣 𐥤 𐥥 𐥦 𐥧 𐥨 𐥩 𐥪 𐥫 𐥬 𐥭 𐥮 𐥯 𐥰 𐥱 𐥲 𐥳 𐥴 𐥵 𐥶 𐥷 𐥸 𐥹 𐥺 𐥻 𐥼 𐥽 𐥾 𐥿 𐦀 𐦁 𐦂 𐦃 𐦄 𐦅 𐦆 𐦇 𐦈 𐦉 𐦊 𐦋 𐦌 𐦍 𐦎 𐦏 𐦐 𐦑 𐦒 𐦓 𐦔 𐦕 𐦖 𐦗 𐦘 𐦙 𐦚 𐦛 𐦜 𐦝 𐦞 𐦟 𐦠 𐦡 𐦢 𐦣 𐦤 𐦥 𐦦 𐦧 𐦨 𐦩 𐦪 𐦫 𐦬 𐦭 𐦮 𐦯 𐦰 𐦱 𐦲 𐦳 𐦴 𐦵 𐦶 𐦷 𐦸 𐦹 𐦺 𐦻 𐦼 𐦽 𐦾 𐦿 𐧀 𐧁 𐧂 𐧃 𐧄 𐧅 𐧆 𐧇 𐧈 𐧉 𐧊 𐧋 𐧌 𐧍 𐧎 𐧏 𐧐 𐧑 𐧒 𐧓 𐧔 𐧕 𐧖 𐧗 𐧘 𐧙 𐧚 𐧛 𐧜 𐧝 𐧞 𐧟 𐧠 𐧡 𐧢 𐧣 𐧤 𐧥 𐧦 𐧧 𐧨 𐧩 𐧪 𐧫 𐧬 𐧭 𐧮 𐧯 𐧰 𐧱 𐧲 𐧳 𐧴 𐧵 𐧶 𐧷 𐧸 𐧹 𐧺 𐧻 𐧼 𐧽 𐧾 𐧿 𐨀 𐨁 𐨂 𐨃 𐨄 𐨅 𐨆 𐨇 𐨈 𐨉 𐨊 𐨋 𐨌 𐨍 𐨎 𐨏 𐨐 𐨑 𐨒 𐨓 𐨔 𐨕 𐨖 𐨗 𐨘 𐨙 𐨚 𐨛 𐨜 𐨝 𐨞 𐨟 𐨠 𐨡 𐨢 𐨣 𐨤 𐨥 𐨦 𐨧 𐨨 𐨩 𐨪 𐨫 𐨬 𐨭 𐨮 𐨯 𐨰 𐨱 𐨲 𐨳 𐨴 𐨵 𐨶 𐨷 𐨸 𐨹 𐨺 𐨻 𐨼 𐨽 𐨾 𐨿 𐩀 𐩁 𐩂 𐩃 𐩄 𐩅 𐩆 𐩇 𐩈 𐩉 𐩊 𐩋 𐩌 𐩍 𐩎 𐩏 𐩐 𐩑 𐩒 𐩓 𐩔 𐩕 𐩖 𐩗 𐩘 𐩙 𐩚 𐩛 𐩜 𐩝 𐩞 𐩟 𐩠 𐩡 𐩢 𐩣 𐩤 𐩥 𐩦 𐩧 𐩨 𐩩 𐩪 𐩫 𐩬 𐩭 𐩮 𐩯 𐩰 𐩱 𐩲 𐩳 𐩴 𐩵 𐩶 𐩷 𐩸 𐩹 𐩺 𐩻 𐩼 𐩽 𐩾 𐩿 𐪀 𐪁 𐪂 𐪃 𐪄 𐪅 𐪆 𐪇 𐪈 𐪉 𐪊 𐪋 𐪌 𐪍 𐪎 𐪏 𐪐 𐪑 𐪒 𐪓 𐪔 𐪕 𐪖 𐪗 𐪘 𐪙 𐪚 𐪛 𐪜 𐪝 𐪞 𐪟 𐪠 𐪡 𐪢 𐪣 𐪤 𐪥 𐪦 𐪧 𐪨 𐪩 𐪪 𐪫 𐪬 𐪭 𐪮 𐪯 𐪰 𐪱 𐪲 𐪳 𐪴 𐪵 𐪶 𐪷 𐪸 𐪹 𐪺 𐪻 𐪼 𐪽 𐪾 𐪿 𐫀 𐫁 𐫂 𐫃 𐫄 𐫅 𐫆 𐫇 𐫈 𐫉 𐫊 𐫋 𐫌 𐫍 𐫎 𐫏 𐫐 𐫑 𐫒 𐫓 𐫔 𐫕 𐫖 𐫗 𐫘 𐫙 𐫚 𐫛 𐫜 𐫝 𐫞 𐫟 𐫠 𐫡 𐫢 𐫣 𐫤 𐫥 𐫦 𐫧 𐫨 𐫩 𐫪 𐫫 𐫬 𐫭 𐫮 𐫯 𐫰 𐫱 𐫲 𐫳 𐫴 𐫵 𐫶 𐫷 𐫸 𐫹 𐫺 𐫻 𐫼 𐫽 𐫾 𐫿 𐬀 𐬁 𐬂 𐬃 𐬄 𐬅 𐬆 𐬇 𐬈 𐬉 𐬊 𐬋 𐬌 𐬍 𐬎 𐬏 𐬐 𐬑 𐬒 𐬓 𐬔 𐬕 𐬖 𐬗 𐬘 𐬙 𐬚 𐬛 𐬜 𐬝 𐬞 𐬟 𐬠 𐬡 𐬢 𐬣 𐬤 𐬥 𐬦 𐬧 𐬨 𐬩 𐬪 𐬫 𐬬 𐬭 𐬮 𐬯 𐬰 𐬱 𐬲 𐬳 𐬴 𐬵 𐬶 𐬷 𐬸 𐬹 𐬺 𐬻 𐬼 𐬽 𐬾 𐬿 𐭀 𐭁 𐭂 𐭃 𐭄 𐭅 𐭆 𐭇 𐭈 𐭉 𐭊 𐭋 𐭌 𐭍 𐭎 𐭏 𐭐 𐭑 𐭒 𐭓 𐭔 𐭕 𐭖 𐭗 𐭘 𐭙 𐭚 𐭛 𐭜 𐭝 𐭞 𐭟 𐭠 𐭡 𐭢 𐭣 𐭤 𐭥 𐭦 𐭧 𐭨 𐭩 𐭪 𐭫 𐭬 𐭭 𐭮 𐭯 𐭰 𐭱 𐭲 𐭳 𐭴 𐭵 𐭶 𐭷 𐭸 𐭹 𐭺 𐭻 𐭼 𐭽 𐭾 𐭿 𐮀 𐮁 𐮂 𐮃 𐮄 𐮅 𐮆 𐮇 𐮈 𐮉 𐮊 𐮋 𐮌 𐮍 𐮎 𐮏 𐮐 𐮑 𐮒 𐮓 𐮔 𐮕 𐮖 𐮗 𐮘 𐮙 𐮚 𐮛 𐮜 𐮝 𐮞 𐮟 𐮠 𐮡 𐮢 𐮣 𐮤 𐮥 𐮦 𐮧 𐮨 𐮩 𐮪 𐮫 𐮬 𐮭 𐮮 𐮯 𐮰 𐮱 𐮲 𐮳 𐮴 𐮵 𐮶 𐮷 𐮸 𐮹 𐮺 𐮻 𐮼 𐮽 𐮾 𐮿 𐯀 𐯁 𐯂 𐯃 𐯄 𐯅 𐯆 𐯇 𐯈 𐯉 𐯊 𐯋 𐯌 𐯍 𐯎 𐯏 𐯐 𐯑 𐯒 𐯓 𐯔 𐯕 𐯖 𐯗 𐯘 𐯙 𐯚 𐯛 𐯜 𐯝 𐯞 𐯟 𐯠 𐯡 𐯢 𐯣 𐯤 𐯥 𐯦 𐯧 𐯨 𐯩 𐯪 𐯫 𐯬 𐯭 𐯮 𐯯 𐯰 𐯱 𐯲 𐯳 𐯴 𐯵 𐯶 𐯷 𐯸 𐯹 𐯺 𐯻 𐯼 𐯽 𐯾 𐯿 𐰀 𐰁 𐰂 𐰃 𐰄 𐰅 𐰆 𐰇 𐰈 𐰉 𐰊 𐰋 𐰌 𐰍 𐰎 𐰏 𐰐 𐰑 𐰒 𐰓 𐰔 𐰕 𐰖 𐰗 𐰘 𐰙 𐰚 𐰛 𐰜 𐰝 𐰞 𐰟 𐰠 𐰡 𐰢 𐰣 𐰤 𐰥 𐰦 𐰧 𐰨 𐰩 𐰪 𐰫 𐰬 𐰭 𐰮 𐰯 𐰰 𐰱

2. Bingo

Program Name: Bingo.java

Input File: bingo.dat

Bingo is a game played with 24 integers in the range [1, 75] placed randomly in a 5 x 5 matrix on a card and a free space placed in the middle square of the card. The five columns are named B, I, N, G, and O respectively from left to right. For a card to be valid, it must meet the following requirements:

B column can only have integers 1 through 15

I column can only have integers 16 through 30

N column can only have integers 31 through 45

G column can only have integers 46 through 60

O column can only have integers 61 through 75

There is exactly one free space (marked by an asterisk (*)) and it must be in 3rd square of the N column

B	I	N	G	O
3	17	33	49	64
6	21	44	56	73
14	25	*	59	69
9	16	45	46	61
5	30	37	60	70

You are to write a program that verifies that a given Bingo card meets the requirements above and is therefore a valid Bingo card.

Input

The first line will contain a single integer *n* that indicates the number of Bingo cards to follow. Each of the next *n* sets of 5 lines will contain a prospective Bingo card. Each card will have five integers on each line (except the line that has an asterisk (*) indicating the free space instead of an integer). The integers will be in the range [1,75]. No integer will appear more than once on a given card.

Output

For each card, print `VALID` if the card is a valid card or print `INVALID` if the card is not a valid card.

Example Input File

```
2
1 16 31 46 61
2 17 32 47 62
3 18 * 48 63
4 19 34 49 64
5 20 35 50 65
1 19 35 47 63
12 21 41 46 73
7 15 * 59 61
14 29 34 55 70
2 30 40 60 68
```

Example Output to Screen

```
VALID
INVALID
```

3. Burning Down the Disc

Program Name: Burn.java

Input File: burn.dat

Steve is burning a CD of his favorite music. The problem is that every time he tries to burn a CD, the songs he wants to put on the CD take up more space than is available. Since Steve values all the songs equally (otherwise he wouldn't want to put them on the CD), his goal is to fill the CD with as many minutes of music as possible.

Write a program that will tell Steve which songs to put on the CD so he will have the least number of minutes wasted. No song is to be placed on the CD more than once. It is possible that none of the songs will fit on the CD.

Input

- The first line will contain a single integer n that indicates the number of data sets to follow.
- Each data set will consist of three lines.
 - The first line of each data set will be a single integer c that indicates the capacity, in minutes, available on the CD for this data set. Each data set has only a single CD.
 - The second line of each data set will be a single integer m that indicates how many different songs Steve is considering to put on this CD.
 - The third line of each data set will be m integers, separated by spaces, that indicate how long each song is in minutes.

Output

- Each data set will have a single optimum solution that results in the least amount of wasted space on the CD. For each data set, output a single line that contains the following:
 - The line will start with "CD <num>: " where <num> is the data set number.
 - The next value in the line is an integer that indicates how much space is unused on the CD for the optimum solution, followed by a single space.
 - The rest of the line will contain the length in minutes of the songs to put on the CD in ascending order and separated by a space.

Example Input File

```
3
50
6
40 30 22 22 10 5
24
5
30 40 35 30 25
50
10
40 60 22 15 51 22 5 55 32 70
```

Example Output To Screen

```
CD 1: 0 10 40
CD 2: 24
CD 3: 1 5 22 22
```

4. Echo Print

Program Name: Echo.java

Input File: echo.dat

Virgil is writing a program that uses an input file. He wants to print the input file to check that he is reading it correctly. Write a program that will read one line at a time and then print that line to the screen.

Input

The input file contains an unknown number of lines.

Output

Print all of the lines in the input file.

Example Input File

```
Mary had a little lamb  
Jack be nimble, Jack be quick  
Hickory, Dickery, Dock
```

Example Output To Screen

```
Mary had a little lamb  
Jack be nimble, Jack be quick  
Hickory, Dickery, Dock
```

5. Palindromic Square

Program Name: Palindromic.java

Input File: palindromic.dat

A palindromic integer reads the same forwards or backwards. All single digit integers are palindromic. But some palindromic integers are more interesting than others. For example, the palindromic integer 595 can be expressed as the sum of the squares of consecutive integers: $595 = 6^2 + 7^2 + 8^2 + 9^2 + 10^2 + 11^2 + 12^2$. We will call an integer a palindromic square if it is a palindromic integer that can be expressed as the sum of the squares of two or more consecutive positive integers.

You will be given a set of integers to determine whether each integer is first a palindromic integer, and if so, whether the integer is also a palindromic square.

Input

The first line will contain a single integer n that indicates the number of lines to follow. Each of the next n lines will contain a single integer to be tested.

Output

Print each integer followed by a space, whether the integer is PALINDROMIC or NOT PALINDROMIC followed by a space, and finally, for palindromic integers only, whether the palindromic integer is SQUARE or NOT SQUARE.

Example Input File

```
4
5
121
595
818
```

Example Output to Screen

```
5 PALINDROMIC SQUARE
121 PALINDROMIC NOT SQUARE
595 PALINDROMIC SQUARE
818 PALINDROMIC SQUARE
```

6. Shiffles

Program Name: Shiffles.java

Input File: shiffles.dat

Two words are said to be Shiffles if the letters in one word are shifted some number of letters to the right in the alphabet, then the letters either form a new word or may be rearranged to form a new word.

For example, each of the letters in the word `FIB` are shifted three letters to the right so `F` shifts to `I`, `I` shifts to `L` and `B` shifts to `E`. The new letters, `ILE` are then rearranged to form the word `LIE`.

You will be given two words to determine if the second word is a “shuffle” of the first word.

Input

The first line will contain a single integer `n` that indicates the number of lines to follow. Each of the next `n` lines will contain an integer `m` that will indicate the number of letters for the shift, a space, and then two words separated by a space.

Assumptions

Both words will consist of the same number of uppercase letters. The last `m` letters of the alphabet will not appear in the first word.

Output

Print `YES` if the second word is a “shuffle” of the first word or print `NO` if it is not.

Example Input File

```
3
3 FIB LIE
5 BAD FOG
7 NILE PLUS
```

Example Output to Screen

```
YES
NO
YES
```