

Problem 1: Cards Against Programmability

5 Points

Problem ID: `cards`

Rank: 1

Introduction

Building a house of cards in real life requires cards, patience, and agility—three things that the average programmer doesn't have. We can circumvent these puny physical limitations by making houses of cards digitally!

Your task is to create a program that will display a house of cards with a given number of layers.

Program Input

The first line of the file `cards.in.txt` will contain a positive integer T denoting the number of test cases that follow. Each test case will have the following input:

- A single positive integer n denoting the number of layers the house of cards must have.

Example Input:

```
3
3
1
5
```

Program Output

For each test case, your program should output the corresponding deck of cards based on the following criteria:

- A layer in a house of cards counts as a single row of cards, displayed with alternating slashes / and \ to represent peaks.
- The top row of the house of cards must consist of a single apex /\
- Each layer in the house of cards must have one greater apex than the row above it.
- Each test case output should be separated by a blank line.

Example Output:

 \wedge

Problem Constraints

$$T \leq 100$$

$$0 \leq n \leq 100$$