UVa Online Judge

110 Meta-Loopless Sorts

Sorting holds an important place in computer science. Analyzing and implementing various sorting algorithms forms an important part of the education of most computer scientists, and sorting accounts for a significant percentage of the world's computational resources. Sorting algorithms range from the bewilderingly popular Bubble sort, to Quicksort, to parallel sorting algorithms and sorting networks. In this problem you will be writing a program that creates a sorting program (a meta-sorter).

The problem is to create several programs whose output is a standard Pascal programs that sorts n numbers where n is the only input to the program you will write. The Pascal program generated by your program must have the following properties:

- They must begin with `program sort(input,output);`
- They must declare storage for exactly n `integer` variables. The names of the variables must come from the first n letters of the alphabet (a,b,c,d,e,f).
- A single `readln` statement must read in values for all the integer variables.
- Other than `writeln` statements, the only statements in the program are `if then else` statements. The boolean conditional for each `if` statement must consist of one strict inequality (either `<` or `>`) of two integer variables. Exactly $n!$ `writeln` statements must appear in the program.
- Exactly three semi-colons must appear in the programs
  1. after the program header: `program sort(input,output);`
  2. after the variable declaration: `... : integer;`
  3. after the `readln` statement: `readln(...);`
- No redundant comparisons of integer variables should be made. For example, during program execution, once it is determined that $a < b$, variables a and b should not be compared again.
- Every `writeln` statement must appear on a line by itself.
- The programs must compile. Executing the program with input consisting of any arrangement of any n distinct integer values should result in the input values being printed in sorted order.

For those unfamiliar with Pascal syntax, the example at the end of this problem completely defines the small subset of Pascal needed.

Input

The input consist on a number in the first line indicating the number M of programs to make, followed by a blank line. Then there are M test cases, each one consisting on a single integer n on a line by itself with $1 \leq n \leq 8$.

There will be a blank line between test cases.

Output

The output is M compilable standard Pascal programs meeting the criteria specified above.

Print a blank line between two consecutive programs.

Sample Input

```
1

3
```

Sample Output

```
program sort(input,output);
var
a,b,c : integer;
begin
  readln(a,b,c);
  if a < b then
    if b < c then
      writeln(a,b,c)
    else if a < c then
      writeln(a,c,b)
    else
      writeln(c,a,b)
  else
    if a < c then
      writeln(b,a,c)
    else if b < c then
      writeln(b,c,a)
    else
      writeln(c,b,a)
end.
```