(See https://cs.stanford.edu/~knuth/programs.html for date.)

1. Intro. This simple program calculates Schensted's Y function. Consider the array

```
    x
   x x        x
  x o o      x o      x          .
 o o x x    o o x    o o    o
x o x x x  o o x x  o o x  o o  o
```

The first nine columns of these five rows were given as standard input; this array shows the standard output.

In general the standard input should consist of $n+1$ lines of $2n+1$ characters, for some n, using only spaces and `x`'s and `o`'s. (Otherwise who knows what might occur. I wrote this in a terrific hurry.)

```
#define maxn 100
#include <stdio.h>
  char a[maxn + 1][maxn + 1][maxn + maxn + 1];
  main( )
  {
    register int i, j, k, n, s;
    ⟨Read the input into a[0], determining n 2⟩;
    for (k = 1; k ≤ n; k++) ⟨Compute a[k] from a[k − 1] 3⟩;
    ⟨Print the results 4⟩;
  }
```

2. ⟨Read the input into a[0], determining n 2⟩ ≡

```
  fgets(a[0][0], maxn + 2, stdin);
  for (n = 0; a[0][0][n] ≡ '␣'; n++) ;
  a[0][0][n + n + 1] = '\0';
  for (k = 1; k ≤ n; k++) {
    fgets(a[0][k], maxn + 2, stdin);
    a[0][k][n + n + 1] = 0;
  }
```

This code is used in section 1.

3. ⟨Compute $a[k]$ from $a[k-1]$ 3⟩ ≡

```
  for (j = 0; j ≤ n − k; j++) {
    for (i = 0; i ≤ n + n − k − k; i++) a[k][j][i] = '␣';
    for (i = n − k − j; i ≤ n − k + j; i += 2) {
      s = 0;
      if (a[k − 1][j][i + 1] ≡ 'o') s++;
      if (a[k − 1][j + 1][i] ≡ 'o') s++;
      if (a[k − 1][j + 1][i + 2] ≡ 'o') s++;
      a[k][j][i] = (s > 1 ? 'o' : 'x');
    }
  }
```

This code is used in section 1.

4. ⟨Print the results 4⟩ ≡

```
  for (k = 0; k ≤ n; k++) {
    printf(a[0][k]);
    for (j = 1; j ≤ k; j++) printf("␣␣%s", a[j][k − j]);
    printf("\n");
  }
```

This code is used in section 1.

5. Index.

⟨ Compute $a[k]$ from $a[k-1]$ 3 ⟩ Used in section 1.
⟨ Print the results 4 ⟩ Used in section 1.
⟨ Read the input into a[0], determining n 2 ⟩ Used in section 1.

YPLAY