

Copyright © 1996 William B. Ackerman

SDTTY DEMO

Here is a demonstration of the use of `sdtty`, illustrating various concepts and other capabilities. You should follow along on your computer. You must use `sdtty` version 31.67 or later.

We will write a long and horrible sequence that illustrates a number of things that the program can do, and shows how you can get it to do them.

First, we start the program. We will run it at C2.

If using DOS or UNIX, type `sdtty` and press `(ENTER)` at the prompt:

```
C:\SD> sdtty (ENTER)
```

If using Windows, click on the `sdtty` icon. In either case, you should see:

```
Enter the level:
```

The program wants us to type the level. Type `c2` and press `(ENTER)`:

```
Enter the level: c2 (ENTER)
Sd: reading database.....done
SD -- square dance caller's helper.
```

```
Copyright (c) 1991-1996 William B. Ackerman
and Stephen Gildea.
```

```
SD comes with ABSOLUTELY NO WARRANTY;
for details see the license.
```

```
This is free software, and you are
welcome to redistribute it.
```

```
At any time that you don't know what you can type,
type a question mark (?). The program will show you all
legal choices.
```

```
Version Sd 31.67 : db31.67 : ui1.8tty      C2
Output file is "sequence.C2"
Enter startup command>
```

After displaying all the introductory stuff, the program wants us to enter a startup command. The line

```
Enter startup command>
```

is called a *prompt*. If we type a question mark at this point, it will list the possible things we could type. Here is what it displays:

```

Enter startup command> ?
exit from the program
heads 1p2p
sides 1p2p
heads start
sides start
just as they are
toggle concept levels
toggle active phantoms
change output file
change title

```

```
Enter startup command>
```

It displays a number of choices, and then gives us another prompt. We will type *heads start*, and press `(ENTER)`:

```
Enter startup command> heads start (ENTER)
Sd 31.67 : db31.67 : ui1.8tty      C2
```

```
4B>  3GV  3BV  2G<
```

```
4G>  1B^  1G^  2B<
```

```
1:  HEADS
```

```
-->
```

It displays some bookkeeping information, and shows us the setup with the heads in the middle, waiting for a call. Below that, it shows the current line of our sequence. It is line 1, and all it knows so far is that it is going to start with *heads*.

The `-->` prompt is the general prompt that it gives when it wants us to enter a call.

If we were to type question mark at this point, it would show over 500 possible calls, concepts, and other commands, so we won't do it for this demo (though you may wish to try it).

We type *pass the ocean*, and press `(ENTER)`.

```
--> pass the ocean (ENTER)
1:  HEADS pass the ocean
```

```
1B>
```

```
4B>  1G<  2G<
```

```
4G>  3G>  2B<
```

```
3B<
```

```
2:
```

```
(diamond/qtag)-->
```

It has now done the call. The first line of the sequence is now complete: *HEADS pass the ocean*. This line will be printed on the finished sequence. It shows the result setup,

shows that line 2 has nothing on it yet, and gives us another prompt. The time, the prompt reminds us that we are in diamonds or quarter tags.

We type *extend* and press `ENTER`.

```
(diamond/qtag)--> extend ENTER
Sd 31.67 : db31.67 : ui1.8tty      C2

1:  HEADS pass the ocean

      1B>
4B>  1G<  2G<
4G>  3G>  2B<
      3B<

2:  extend

4B>  1B>

1G<  2G<

4G>  3G>

3B<  2B<

3:
(waves)-->
```

It shows the full sequence that we have so far. After it does any call, it shows a picture of the setup before and after. In this case, it has done the *extend* on line 2, and is showing the setup before and after. Those pictures will not be printed on the final sequence (unless we request that the program do so, by giving the *keep picture* command). The pictures before and after the last call are displayed on the screen just so we can be sure that we agree with what the computer did.

The prompt shows that we are in waves.

We want to do a tandem swing thru at this point. Tandem is a concept.

There are two ways of dealing with concepts. One is to type just the concept name, press `ENTER`, type the call name, and press `ENTER` again. This method is less convenient, but it always works.

The more convenient way is simply to type the concept(s) and call all on one line, followed by `ENTER`, as in *tandem swing thru* `ENTER`.

It won't make any difference in the final printout which method you used.

The significance of this is discussed in more detail in the "Entering Concepts" chapter of the reference manual. For the purposes of this demo, we will use the one-line method wherever possible. So we type *tandem swing thru*, and press `ENTER`.

```

(waves)--> tandem swing thru ENTER
1:  HEADS pass the ocean
2:  extend

4B>  1B>

1G<  2G<

4G>  3G>

3B<  2B<

3:  TANDEM swing thru

2G>  1G>

3B<  2B<

4B>  1B>

3G<  4G<

      resolve is:
extend, right and left grand (1/4 promenade)
4:
(waves)-->

```

It is now showing us 3 lines. There are pictures of the setup before and after the tandem swing thru, but it is not showing earlier pictures. It assumes we are satisfied with what went on previously.

Line 4 is now blank and ready to be filled in.

The program has noticed that we happen to be in a setup from which we could say **extend, right and left grand**, and is pointing that out to us. If we wanted to end the sequence right here, we could do so (by typing *write this sequence* or pressing F10). Instead, we are going to continue, and forget all about that resolve.

Let's demonstrate some concepts that identify people. We will do a 1/2 acey deucey to make a 3x1 diamond setup.

Type *1/2 acey deucey*, and press ENTER.

```

--> 1/2 acey deucey ENTER
2:   extend
3:   TANDEM swing thru

2G>  1G>

3B<  2B<

4B>  1B>

3G<  4G<

4:   1/2, acey deucey

           2G>

3G^  3BV  4B^  2BV  1B^  1GV

           4G<

5:
-->

```

The program puts commas in the final sequence where needed for clarity. You never type them in.

It now shows us the full line 4, and the 3x1 diamond that resulted. It places a comma after some concepts to make everything completely unambiguous. You never need to type the commas.

As the sequence progresses, early parts of it may scroll off the top of the screen and be invisible. Note that, in this example, we have not shown line 1, but are only showing the last few calls. How much you see on your screen may be different. Don't worry, the computer remembers the whole sequence and will print it all out when we are done.

We would like the girls to circulate in the big diamond around the outside. We must use the **disconnected** concept to do this. The disconnected concept is C2. Now it happens that you could almost certainly get dancers to do what you want at lower levels by saying **girls diamond circulate around the outside**, or something like that, but the program doesn't know that. It insists on meticulously correct use of concepts. If we wanted to do this at a level below C2, we would therefore have to tell the program that we really want to use a concept that isn't legitimate. We can do this by giving the **toggle concept levels** command. But in the present situation we don't need to do that, because we are running the program at C2. So we use the **girls disconnected** concept. There are **<anyone> disconnected** concepts for a number of designators, such as boys, girls, heads, sides, centers, ends, beaus, belles, etc. Many of them would of course be illegal in the present context, but **girls disconnected** is legal.

```

--> girls disconnected diamond circulate ENTER
3: TANDEM swing thru
4: 1/2, acey deucey

      2G>

3G^ 3BV 4B^ 2BV 1B^ 1GV

      4G<

5: GIRLS DISCONNECTED diamond circulate

      3G>

4G^ 3BV 4B^ 2BV 1B^ 2GV

      1G<

6:
-->

```

The program shows a few calls back, and the just-completed call with pictures before and after. We can use other designators also. Let's do a **heads disconnected diamond circulate** at this point. As usual, we will type the concept and the call, and press ENTER.

The program responds by showing this:

```

--> heads disconnected diamond circulate ENTER
4: 1/2, acey deucey
5: GIRLS DISCONNECTED diamond circulate

      3G>

4G^ 3BV 4B^ 2BV 1B^ 2GV

      1G<

6: HEADS DISCONNECTED diamond circulate

      1B<

4G^ 1G^ 4B^ 2BV 3GV 2GV

      3B>

7:
-->

```

We could have the boys diamond circulate. They are not disconnected, so we just use the **boys** concept. As with disconnected, there are **boys, girls, centers, ends, leads,**

trailers, etc. concepts. From now on, we will usually not describe in detail the act of entering concepts and calls.

```
--> boys diamond circulate ENTER
5:  GIRLS DISCONNECTED diamond circulate
6:  HEADS DISCONNECTED diamond circulate
```

1B<

4G^ 1G^ 4B^ 2BV 3GV 2GV

3B>

7: BOYS diamond circulate

4B>

4G^ 1G^ 1BV 3B^ 3GV 2GV

2B<

We could also have designated them as the centers, since the centers of a 3x1 diamond is the center diamond. Let's try it. But first, let's undo the `boys diamond circulate` that we just entered. The `undo` command does this. There are two ways of entering this command. One way, of course, is to type `undo`, and press ENTER. (The full name of the command is `undo last call`, but we just need to type `undo`. The program will show the full name.) The other way is to press function key F9. Many function keys are arranged to perform common operations. The list is given in the "Terminal Interface" chapter of the reference manual.

```
--> undo last call ENTER
5:  GIRLS DISCONNECTED diamond circulate
```

3G>

4G^ 3BV 4B^ 2BV 1B^ 2GV

1G<

6: HEADS DISCONNECTED diamond circulate

1B<

4G^ 1G^ 4B^ 2BV 3GV 2GV

3B>

7:

-->

The program has erased the boys *diamond circulate*, and is showing us the last few calls of the remaining sequence. It is prompting us for the new line 7. Have the centers *diamond circulate*, by typing *centers diamond circulate*, and pressing ENTER.

```
--> centers diamond circulate ENTER
5:  GIRLS DISCONNECTED diamond circulate
6:  HEADS DISCONNECTED diamond circulate

      1B<

4G^  1G^  4B^  2BV  3GV  2GV

      3B>

7:  CENTERS diamond circulate

      4B>

4G^  1G^  1BV  3B^  3GV  2GV

      2B<
```

There is also a *center diamond* concept. So let's undo the last call, either by typing the command name or by pressing F9, and try it.

```
--> F9
5:  GIRLS DISCONNECTED diamond circulate

      3G>

4G^  3BV  4B^  2BV  1B^  2GV

      1G<

6:  HEADS DISCONNECTED diamond circulate

      1B<

4G^  1G^  4B^  2BV  3GV  2GV

      3B>

7:
```



```

--> center diamond diamond circulate ENTER
5:  GIRLS DISCONNECTED diamond circulate
6:  HEADS DISCONNECTED diamond circulate

      1B<

4G^  1G^  4B^  2BV  3GV  2GV

      3B>

7:  CENTER DIAMOND diamond circulate

      4B>

4G^  1G^  1BV  3B^  3GV  2GV

      2B<

8:
-->

```

Now we would like to have the heads in the center wave do a swing thru. We can't say **centers**, because the centers of a 3x1 diamond are the center diamond, not the center line. Instead, we use the *center 1x4* concept.

```

      4B>

4G^  1G^  1BV  3B^  3GV  2GV

      2B<

8:  CENTER 1X4 swing thru

      4B>

4G^  1B^  3GV  1G^  3BV  2GV

      2B<

```

The concept is called *center 1x4* instead of *center line* or *center wave* or *center column* because otherwise it would lead to a proliferation of concept names that basically do the same thing. When reading the card, you should probably say "center wave of 4," or something like that.

Of course, instead of *center 1x4*, we could have designated the people directly, since they are the heads. Let's have them trade the wave to demonstrate that:

4B>

4G^ 1B^ 3GV 1G^ 3BV 2GV

2B<

9: HEADS trade the wave

4B>

4G^ 1GV 3B^ 1BV 3G^ 2GV

2B<

Now we would like to have the center wave of 6 do a grand swing thru. We use the *center 1x6* concept:

4B>

4G^ 1GV 3B^ 1BV 3G^ 2GV

2B<

10: CENTER 1X6 GRAND swing thru

4B>

1G^ 1BV 4G^ 2GV 3B^ 3GV

2B<

Incidentally, *grand* is a concept. So, if we were typing concepts one at a time, we could have typed *center 1x6*, pressed ENTER, *grand*, pressed ENTER, and *swing thru*, and pressed ENTER a third time. Of course, we don't bother with that. Just type *center 1x6 grand swing thru* and press ENTER.

Let's have the centers flip the diamond while the ends explode. For this we use the *centers (while the others)* concept. As usual, there is a whole series of these concepts, for centers, heads, sides, leads, trailers, and so on.

Yes, the concept actually looks like that when we type it in. It even has to have the parentheses. The program will print it out on the card looking much better than that, but, while we type it in, the name of the concept is "*centers (while the others)*".

A very important thing you need to know is that one-line entry is not allowed for those concepts that take two calls. Therefore, you must type the name of the concept, press ENTER, and type each of the two calls, pressing ENTER after each one.

Now, because of the *completion* property of Sdttty, we don't have to type the whole name "*centers (while the others)*". As soon as we have typed the left parenthesis, the program knows what concept it has to be.

One thing we could do is type *centers* (and then press `(ESC)`. The program will put the rest of the concept on the screen just as though we had typed it.

```
--> centers ( (ESC)
the screen now shows
```

```
--> centers (while the others)
```

because the program has typed *while the others*) for us. We can now press `(ENTER)` to tell the program that we want that concept.

Another thing we could have done, if we were really sure we knew what we wanted without seeing it, is to just press `(ENTER)` after typing *centers* (, without pressing `(ESC)`.

Either way, the program takes the *centers (while the others)* concept, and gives us another prompt. This is the prompt for the call that we want the centers to do.

```
--> centers (while the others)
11:  CENTERS
-->
```

Line 11 just says *CENTERS*. It will say more very soon. At the prompt, we type *flip the diamond*, and press `(ENTER)`.

```
--> flip the diamond (ENTER)
11:  CENTERS flip the diamond
ENTER SECOND CALL
-->
```

Line 11 is looking more complete. We are given another prompt, with special instructions. It wants the second call. This is, of course, the call for the ends. Type *explode*, and press `(ENTER)`.

```

ENTER SECOND CALL
--> explode ENTER
  9:  HEADS trade the wave
 10:  CENTER 1X6 GRAND swing thru

          4B>

1G^  1BV  4G^  2GV  3B^  3GV

          2B<

11:  CENTERS flip the diamond WHILE THE ENDS explode
Warning:  Ends should opt for setup parallel to their
original line -- concentric rule does not apply.

          4G>
1G>  4B<  3B<
1B>  2B>  3G<
          2G<

12:
(diamond/qtag)-->

```

Line 11 is finished. As promised, it looks much better than it did when we typed it in. The program knows that the *others* are actually the ends, and it has put the whole thing into a reasonable word order. The program's exact wording may not always be ideal—you are responsible for using the right words when you call the card.

The program has put a warning into the sequence, in case someone mistakenly thinks that the *opposite elongation* rule for the concentric concept somehow applies here. Warnings are printed in a wide variety of situations, if the program thinks that there might be some danger of misinterpretation. In most cases, you should not read the warnings when calling.

(Incidentally, instead of *centers* (*while the others*), we could have said *sides* (*while the others*). If we had done this, the sequence would of course say “SIDES flip the diamond WHILE THE HEADS explode”.)

To prepare for what comes next, do a *chain reaction* and *ah so*.

12: chain reaction

4B> 1B>

2G< 1G<

3G> 4G>

3B< 2B<

13: ah so

3B> 2G> 2B> 1G>

3G< 4B< 4G< 1B<

resolve is:
reverse single file promenade (1/4 promenade)

The program reminds us that we have stumbled into a resolve. We're not interested.

Have the ends hinge, by typing *ends hinge*, and ENTER. (We could have said *heads* instead of *ends*, of course.)

14: ENDS hinge

2G> 2B>
3G^ 3BV 1B^ 1GV
4B< 4G<

The boys have a distorted facing diamond. We can use the *boys in your distorted diamond* concept to have them diamond circulate.

15: BOYS IN YOUR DISTORTED DIAMOND diamond circulate

2G> 1B<
3G^ 4B^ 2BV 1GV
3B> 4G<

Have the centers left touch 1/4.

16: CENTERS LEFT touch 1/4

1BV 2G^
3G^ 4B^ 2BV 1GV
4GV 3B^

Now the girls have a distorted wave, and can swing thru.

17: GIRLS IN YOUR DISTORTED WAVE swing thru

```

      1BV  3G^
4G^ 4B^      2BV 2GV
      1GV  3B^

```

Have the heads trail off, then everyone crossfire.

18: HEADS trail off

```

4G^ 4B^ 3GV 3BV 1B^ 1G^ 2BV 2GV

```

19: crossfire

```

3G^ 4GV 2B^ 1BV
3B^ 4BV 2G^ 1GV

```

Have the leads press ahead to make blocks. Incidentally, when we type *leads press ahead*, the program does not consider “leads” to be a concept. The name of the call is simply *leads press ahead*. The same is true of all the other little press-and-truck calls, like *quarter left*, *quarter right*, *u-turn back*, and *shove off*. Of course, this point is of little concern to us at present. We just type *leads press ahead*.

20: leads press ahead

```

3G^  .  2B^  .
.  4GV  .  1BV
3B^  .  2G^  .
.  4BV  .  1GV

```

We have a diagonal wave of four. We can use the *diagonal wave* concept.

21: DIAGONAL WAVE swing thru

```

4G^  .  2B^  .
.  1GV  .  1BV
3B^  .  3G^  .
.  4BV  .  2GV

```

(We could also have identified the people in that wave, by using the *girls in your diagonal wave* concept.)

There are also two diagonal waves of 3, or two diagonal columns of three, depending on what we want to do. Let’s have them identify columns and do a magic circulate. The name of the concept is *diagonal columns of 3*.

22: DIAGONAL COLUMNS OF 3 MAGIC circulate

```

4G^   .   3G^   .
      .   1BV   .   2BV
4B^   .   3B^   .
      .   1GV   .   2GV

```

Have the girls u-turn back and press ahead.

23: girls U-turn back

```

4GV   .   3GV   .
      .   1BV   .   2BV
4B^   .   3B^   .
      .   1G^   .   2G^

```

24: girls press ahead

```

4GV  1BV  3GV  2BV
4B^  1G^  3B^  2G^

```

For the next item, do *cross the K*, and then have the boys single wheel.

25: cross the K

```

1GV  2GV  4BV  3BV
1B^  2B^  4G^  3G^

```

26: BOYS single wheel

Warning: Check a 'C1 phantom' setup.

```

      .           4B^
1GV  2GV   .     .
      .           3B^

      1BV           .
      .           .   4G^  3G^
      2BV           .

```

Do a siamese trail to a diamond.

27: SIAMESE trail to a diamond

```

      3G<
3BV   2B^
      4G<

      2G>
4BV   1B^
      1G>

```

We now want to unwrap the interlocked diamonds. We can type this in any of three ways. *Interlocked*, like *grand*, *cross*, *magic*, *left*, *single*, *split*, and those sorts of things, is considered by the program to be a concept. As such, we could type *interlocked unwrap the diamonds* `(ENTER)`. We could also type *unwrap the interlocked diamonds* directly, with the word order the way it is supposed to be. A number of calls that take *magic*, *interlocked*, *cross*, and *left* modifiers with unusual word order can be typed in that exact word order. It isn't easy to know which ones, so it is always safe to type the concept name first and then see what happens. No matter how we type it, it gets printed out with the correct word order.

28: unwrap the interlocked diamonds

```

2GV   1B^

2BV   1G^

3GV   4B^

3BV   4G^

```

Let's have the column circulate one and a half times. There is a 1-1/2 concept (in fact, there is a 1-M/N concept for any reasonable value of M and N). There is actually a hyphen (minus sign) in the concept name, to avoid confusion when looking at the card. However, you don't need to type the hyphen. You can just type 1 1/2.

```

--> 1 1/2 column circulate (ENTER)
29:  column circulate 1-1/2

```

```

      1G<
1BV  4B^
2GV  4G^
2BV  3B^
      3G>

```

The program has moved the phrase 1-1/2 to the end when it displays the line, and it will also do so in the printed card. It does the same with the concepts *twice* and *<n> times*. You must still type all concepts before the calls that they modify.

Let's have the boys hinge.

30: BOYS hinge

1G<

4B<

1B>

2GV 4G^

3B<

2B>

3G>

The boys were in a disconnected setup. Why didn't we need to say **boys disconnected hinge**? The reason is that hinge is just a two-person call, and the people actually doing each hinge were right next to each other. If we had instead wanted the boys to do a box circulate, we would have had to say **disconnected**.

Have the center diamond flip the diamond, and then do a wheel and deal and tag your neighbor.

33: tag your neighbor

3G< 2G<

1B> 4B>

2B< 3B<

4G> 1G>

Do a *sets in motion but hold the column*.

34: sets in motion but hold the column

1G<

2G> 2B> 3B> 1B< 4B< 4G<

3G>

The call **sets in motion but hold the column** is simply a call with a very long name. There is no **but hold the column** concept.

We can have the four in the middle do a swing thru. The head boys will step to right hands and begin the call, and then finish with the head girls. We can use the **centers**, or **center diamond**, or **heads** concept. As discussed above, the **centers** in a 3x1 diamond (which is what this is, even though people are facing in unusual directions) are the center diamond.

35: CENTERS swing thru
Warning: Some people step to a wave.

```

          1B<
          1G>
2G> 2B>          4B< 4G<
          3G<
          3B>

```

Let's have the centers mix.

36: CENTERS mix

```

          3G>
          3B<
2G> 2B>          4B< 4G<
          1B>
          1G<

```

This would be a good time to demonstrate the <anyone> are tandem concept. There are such concepts (and for as couples too, of course), for all the usual designators. In this case, we will say sides are tandem.

--> *sides are tandem swing and circle 1/4* ENTER
37: SIDES ARE TANDEM, swing and circle 1/4

```

2G> 2B>
  1G<
1B>          3B<
  3G>
4B< 4G<

```

The program put in a comma for us.

Have the centers extend, all trade circulate, and do another sets in motion but hold the column to demonstrate the next thing.

```

38:  CENTERS extend

    2G>  2B>

    1G<  3B<

    1B>  3G>

    4B<  4G<

39:  trade circulate

    3B<  3G<

    4B>  2G>

    4G<  2B<

    1G>  1B>

40:  sets in motion but hold the column

        1B<

    3G>  4G>  2B>  4B<  2G<  1G<

        3B>

```

We are going to have the center column of 6 do a single file dixie sashay, with the girls in tandem. We need to use the `center 1x6` concept for this. The `center 6` concept would be ambiguous.

```

--> center 1x6 girls are tandem single file dixie sashay ENTER
41:  CENTER 1X6, GIRLS ARE TANDEM, SINGLE FILE dixie
    sashay

        1B<
    1GV 4G^
4BV          2B^
    2GV 3G^
        3B>

```

We didn't type in any of those commas. We should read this line as something like "center column of 6, the girls are in tandem, single file dixie sashay."

Let's squeeze the galaxy and flip the hourglass.

42: squeeze the galaxy

```

      1B<
4GV      1G^
      2BV 4B^
3GV      2G^
      3B>

```

43: flip the hourglass

```

1BV  4G^  1GV  4B^

2BV  3G^  2GV  3B^

```

Just to finish the job, we resolve the sequence. The program will not permit us to write the sequence to the file unless it is resolved. We could notice that slip and rims trade back gives us an allemande left.

44: slip

```

1BV  1G^  4GV  4B^

2BV  2G^  3GV  3B^

```

45: rims trade back

```

1GV  2B^  1BV  4G^

2GV  3B^  4BV  3G^

```

resolve is:

left allemande (1/2 promenade)

46:

(left waves)-->

The program is still prompting us for more. The facts that we have a resolve, and that the sequence is 45 calls long, don't mean anything to the computer. In order to really write this out we have to give the `write this sequence` command. We type *write this sequence*, and press `(ENTER)`. (We could also have pressed function key `(F10)`.)

The program prompts us for a comment.

```

(left waves)--> write this sequence
Type comment for this sequence, if desired.
Enter comment:

```

At the prompt, we type *demo*, and then press `(ENTER)`.

The program then displays

```

Appending to existing file.

```

to let us know that it is writing the sequence to a file, and that there was already a file present under the name 'sequence.C2', and that it is not destroying whatever sequences were already in that file. It then displays the entire card for our perusal and delectation:

```

Wed May 15 21:22:28 1996      Sd31.67:db31.67      C2
      demo

HEADS pass the ocean
extend
TANDEM swing thru
1/2, acey deucey
GIRLS DISCONNECTED diamond circulate
HEADS DISCONNECTED diamond circulate
CENTER DIAMOND diamond circulate
CENTER 1X4 swing thru
HEADS trade the wave
CENTER 1X6 GRAND swing thru
CENTERS flip the diamond WHILE THE ENDS explode
  Warning:  Ends should opt for setup parallel to their
           original line -- concentric rule does not apply.
chain reaction
ah so
ENDS hinge
BOYS IN YOUR DISTORTED DIAMOND diamond circulate
CENTERS LEFT touch 1/4
GIRLS IN YOUR DISTORTED WAVE swing thru
HEADS trail off
crossfire
leads press ahead
DIAGONAL WAVE swing thru
DIAGONAL COLUMNS OF 3 MAGIC circulate
girls U-turn back
girls press ahead
cross the K
BOYS single wheel
  Warning:  Check a 'C1 phantom' setup.
SIAMESE trail to a diamond
unwrap the interlocked diamonds
column circulate 1-1/2
BOYS hinge
CENTER DIAMOND flip the diamond
wheel and deal
tag your neighbor
sets in motion but hold the column
CENTERS swing thru
  Warning:  Some people step to a wave.
CENTERS mix

```

```

SIDES ARE TANDEM, swing and circle 1/4
CENTERS extend
trade circulate
sets in motion but hold the column
CENTER 1X6, GIRLS ARE TANDEM, SINGLE FILE dixie sashay
squeeze the galaxy
flip the hourglass
slip
rims trade back
left allemande (1/2 promenade)

```

Sequence written to "sequence.C2".

Enter startup command>

This sequence is so excessively long that, when you follow this demo on your computer, you very likely won't be able to see it all on your screen, but it's all there.

The program then reassures us that the sequence was written:

```
Sequence written to "sequence.C2".
```

It then prompts us for instructions on starting another sequence:

```
Enter startup command>
```

We could type, for example, *sides start*, and do the next sequence, or we could type *exit*.

The file 'sequence.C2' contains the sequence in all its glory. If we print it out, it would probably take 2 or 3 pages. Don't call this card.