EXAMPLE PROGRAM GIANT ADA REPORT

1. Requirements

To write the word "Ada" vertically down the screen using giant letters made up of strings of "*" characters of the form depicted in Table 1.

							*						
						*		*					
					*				*				
				*						*			
			*	*	*	*	*	*	*	*	*		
		*										*	
	*												*

		*	*	*	*	*	*	*		
		*							*	
		*								*
		*								*
		*								*
		*							*	
		*	*	*	*	*	*	*		

Table 1: *Giant Letter Design*

2. Design

Using a top-down hierarchical decomposition approach we can identify the break-down of operations shown in Table 2.

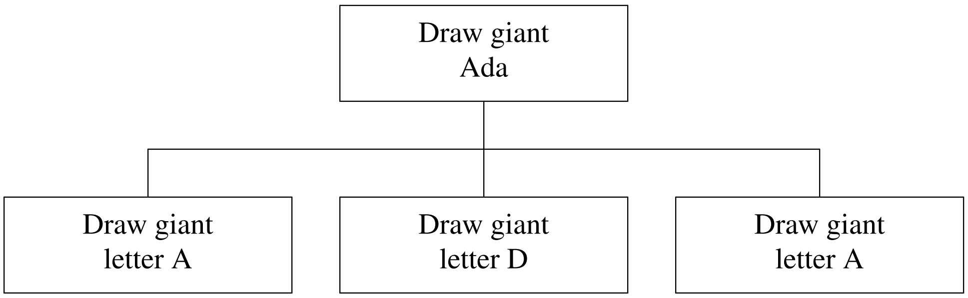
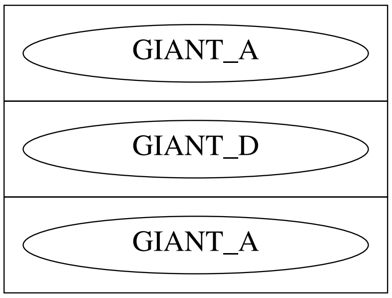


Table 2: *Top-down analysis*

From the above three procedures can be identified:

1) `GIANT_ADA` (top level procedure): Calls to `GIANT_A` and `GIANT_D` procedures.
2) `GIANT_A` (level 2 procedure): Draw giant letter A.
3) `GIANT_D` (level 2 procedure): Draw giant letter D.

The design detail required for each of these is given by the Nassi-Shneiderman charts presented in Table 3. Note that design involves no input, only output.



GIANT_ADA
(Top level procedure)

output "   *"
output "   * *"
output "   *   *"
output "   *   *"
output " *********"
output " *   *"
output "*   *"

GIANT_A PROCEDURE
(Level 2 procedure)

output " ******"
output " *   *"
output " *   *"
output " *   *"
output " *   *"
output " *   *"
output " ******"

GIANT_D PROCEDURE
(Level 2 procedure)

Table 3: *Nassi-Shneiderman Charts*

3. Implementation

Implementation is as shown in table 4.

```
-- GIANT ADA
-- 9 August 1997
-- Frans Coenen
-- Dept Computer Science, University of Liverpool

with CS_IO;
use CS_IO;

procedure GIANT_ADA is

---------------------------------------------------------------------
-- GIANT LETTER A PROCEDURE
        procedure GIANT_A is
        begin
                PUT_LINE("      *");
                PUT_LINE("     * *");
                PUT_LINE("    *   *");
                PUT_LINE("   *     *");
                PUT_LINE("  *********");
                PUT_LINE(" *         *");
                PUT_LINE("*           *");
                NEW_LINE;
        end GIANT_A;
```

Table 4: *Implementation*

```
---------------------------------------------------------------------
-- GIANT LETTER D PROCEDURE
        procedure GIANT_D is
        begin
                PUT_LINE("  *******");
                PUT_LINE("  *      *");
                PUT_LINE("  *       *");
                PUT_LINE("  *       *");
                PUT_LINE("  *       *");
                PUT_LINE("  *      *");
                PUT_LINE("  *******");
                NEW_LINE;
        end GIANT_D;

---------------------------------------------------------------------

-- TOP LEVEL PROCEDURE
begin
        GIANT_A;
        GIANT_D;
        GIANT_A;
end GIANT_ADA;
```

Table 4 Continued

4. Testing

Two level design therefore test each level as it is encoded. In each case, given that there is no input and a straight forward sequence of commands, testing can be carried out through a single execution of the program. A copy of the resulting output is given in Table 5.

```
kuban-428 $ giant_ada
      *
     * *
    *   *
   *     *
  *********
 *         *
*           *

  *******
  *      *
  *       *
  *       *
  *       *
  *      *
  *******

      *
     * *
    *   *
   *     *
  *********
 *         *
*           *
```

Table 5: *Test output*