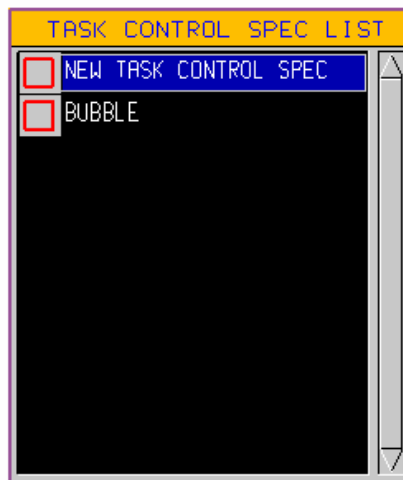
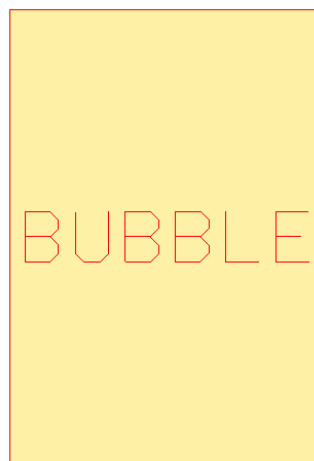


CHAPTER 4: BUBBLE SORT

The next program you will write takes integers as input and sorts them in ascending order using a bubble sort. We shall also look into coding resources and processes in detail. Let's assume the program is already written. When you hit the TASK button the window below opens where you see two Tasks are present. Click on "BUBBLE".



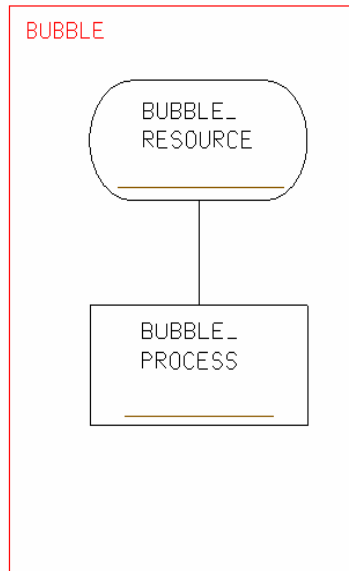
This is the box that will appear on the screen:



The box will need to be uncovered to see the program. At the bottom of the panel there is an 'UNCOVER' button like below:



You can uncover the drawing by first selecting the UNCOVER button, and then clicking on the drawing to be uncovered. After doing this we get the following:



The above window represents the “BUBBLE” Task. It contains one resource, namely the 'BUBBLE_RESOURCE', and one process, 'BUBBLE_PROCESS'. The figure below represents the resource for the BUBBLE program.

A screenshot of a Windows WordPad window titled "BUBBLE_RESOURCE.RES - WordPad". The window has a menu bar with "File", "Edit", "View", "Insert", "Format", and "Help". Below the menu bar is a toolbar with various icons. The main text area contains the following code:

```
***BUBBLE_SORT RESOURCE  
  
VARIABLES  
  1 I           INTEGER  
  1 J           INTEGER  
  1 ARR_SIZE    INTEGER  
  1 TEMP        INTEGER  
  1 ARR         INTEGER QUANTITY(100)
```

At the bottom of the window, there is a status bar that says "For Help, press F1" and a "NUM" button.

Observe that each statement begins on a new line and also different parts of the statement are separated by spaces. VARIABLES is the top-level attribute and it must begin in column 1 through 4. All other resource statements like I and J should begin in column 5

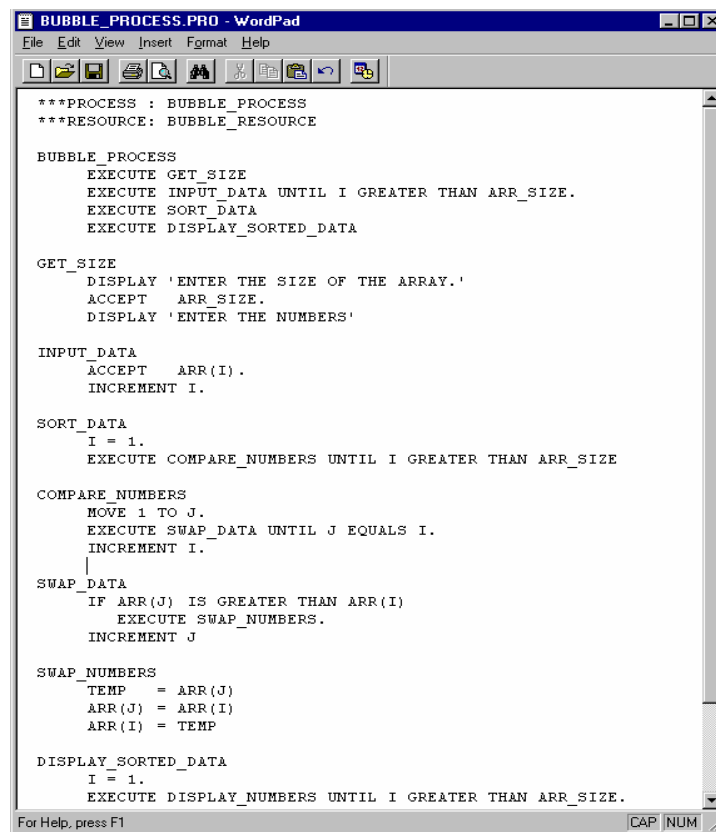
or beyond. A maximum of 72 characters is allowed per line. There are also the level numbers. Level numbers are used to organize a resource into a hierarchical structure of group and elementary attributes. In the BUBBLE_SORT RESOURCE we defined only elementary attributes. The basic subdivisions of a resource are called elementary attributes.

DECLARING AN ARRAY

In order to declare an array, for example ARR in the figure above, the QUANTITY clause is used. It indicates the number of times an attribute or set of attributes with identical structures is repeated. The integer specified within parenthesis (like 10 as in figure above) represents the exact number of occurrences, and must be greater than zero.

INITIAL_VALUE CLAUSE

An INITIAL_VALUE clause allows values to be assigned to resource attributes at the beginning of a task run. For example, if we wanted SIZE to have the initial value 1, we would write INITIAL VALUE 1. The following listing gives the BUBBLE_PROCESS:



```
***PROCESS : BUBBLE_PROCESS
***RESOURCE: BUBBLE_RESOURCE

BUBBLE_PROCESS
EXECUTE GET_SIZE
EXECUTE INPUT_DATA UNTIL I GREATER THAN ARR_SIZE.
EXECUTE SORT_DATA
EXECUTE DISPLAY_SORTED_DATA

GET_SIZE
DISPLAY 'ENTER THE SIZE OF THE ARRAY.'
ACCEPT ARR_SIZE.
DISPLAY 'ENTER THE NUMBERS'

INPUT_DATA
ACCEPT ARR(I).
INCREMENT I.

SORT_DATA
I = 1.
EXECUTE COMPARE_NUMBERS UNTIL I GREATER THAN ARR_SIZE

COMPARE_NUMBERS
MOVE I TO J.
EXECUTE SWAP_DATA UNTIL J EQUALS I.
INCREMENT I.
|

SWAP_DATA
IF ARR(J) IS GREATER THAN ARR(I)
EXECUTE SWAP_NUMBERS.
INCREMENT J

SWAP_NUMBERS
TEMP = ARR(J)
ARR(J) = ARR(I)
ARR(I) = TEMP

DISPLAY_SORTED_DATA
I = 1.
EXECUTE DISPLAY_NUMBERS UNTIL I GREATER THAN ARR_SIZE.
```

In the process, we initially started with two commented lines. The first three stars before the words “PROCESS NAME” and “RESOURCES” represent comments.

RULES

The process BUBBLE_PROCESS consists of different rules (for example, BUBBLE_SORT_PROCESS, DETERMINE_SIZE) each with a unique name. The rule name can be any VSE word (other than reserved words).

Observe that the internal process description consists of statements, which are grouped into rules. Statements and rules form a hierarchical structure for describing processes. Each rule name must appear on a separate line followed by statements, which make up the rule. Observe that each statement begins on a new line in the above figure. And also all process statements must begin in column 5 or beyond, and all rule names must begin in columns 1 through 4. The maximum line length is 72 characters.

EXECUTE STATEMENT

The EXECUTE statement is used to depart from the normal sequence within a rule in order to execute another rule a specified number of times, or until a predetermined condition is satisfied. Below the rule "INPUT_DATA" is executed until I is greater than ARR_SIZE. So whenever an EXECUTE statement is executed, control is transferred to the first statement of the rule named "rule name". And control is always returned to the statement immediately following the EXECUTE statement. Here control is returned to "EXECUTE SORT_DATA" after INPUT_DATA rule is executed.

```
'EXECUTE INPUT_DATA UNTIL I GREATER THAN ARR_SIZE.'
```

ASSIGNING VALUES

Different forms of the assignment statements are used for different types of attributes. In the above example we used only numeric assignments. For numeric attributes as shown we use a numeric assignment or a MOVE statement. Below is an assignment statement. More than one attribute may appear to the left of the assignment statement.

```
'I = 1.'
```

Here we used a "MOVE" statement to assign 1 to J.

```
'MOVE 1 TO J.'
```

The "MOVE" statement is particularly useful when assigning values to a group attribute, which may contain a mixture of attribute types. Note that the movement of nonnumeric data to numeric attributes is not allowed.

INCREMENT STATEMENT

Increment and decrement operations exist for the common operation of adding or subtracting a positive constant value. When the INCREMENT or DECREMENT keywords are used, the value to be added to or subtracted from the named attribute may be omitted. If so, the default is an increment of 1 as shown below. Here "I" is incremented by 1.

```
' INCREMENT I . '
```

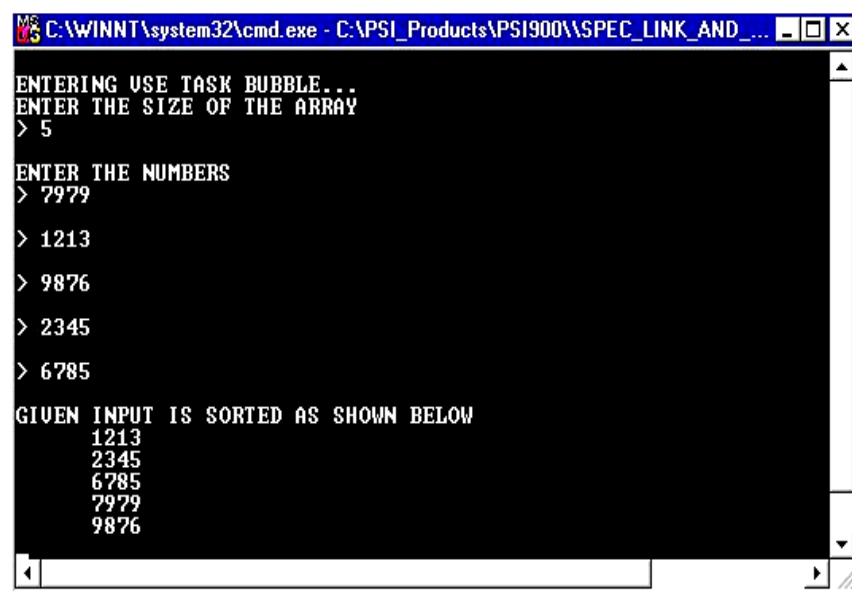
IF STATEMENT

An IF statement must end with a period. The only exception to this rule arises when IF statements are nested, in which case the nested statement must end with a period.

```
' IF ARR(J) IS GREATER THAN ARR(I)  
  EXECUTE SWAP_NUMBERS . '
```

Here if the condition is “true”, the "SWAP_NUMBERS" rule is executed. If a condition is false, the ELSE clause (if present) is executed and then the control passes to the statement following the period.

A sample run of the sorting program is given below:



```
C:\WINNT\system32\cmd.exe - C:\PSI_Products\PSI900\SPEC_LINK_AND_...  
ENTERING USE TASK BUBBLE...  
ENTER THE SIZE OF THE ARRAY  
> 5  
  
ENTER THE NUMBERS  
> 7979  
> 1213  
> 9876  
> 2345  
> 6785  
  
GIVEN INPUT IS SORTED AS SHOWN BELOW  
1213  
2345  
6785  
7979  
9876
```