


EXERCISE 1



Write Number Function

`write_number (5, "this is record number")`

this is record number 1
this is record number 2
this is record number 3
this is record number 4
this is record number 5

Objectives:

- Write a function to open an external file called `records.out`.
- Write a given number of records to the file.



Exercise Description:

The function, `write_number` (`max_number`, `leading_text`) will have two parameters, an integer for the number of records, and a string of text to identify the records. You will create a unique file, and write a unique line for each record. Finally, send a message to the command window informing the user that the execution of the function is completed.

The test case for this function is:

For number of records = 5 and identifier “this is record number” the output file should contain:

```
this is record number 1
this is record number 2
this is record number 3
this is record number 4
this is record number 5
```

Files:

All the files that used in this exercise are listed below. Each list includes the file, where it originated, and a summary of information of how it relates to the exercise.

File	Supplied/Create	Description
<code>write_number.pcl</code>	Create	Create a file that contains a PCL function.
<code>records.out.01</code>	Created	This file is created when the PCL function is executed.

Exercise Procedure:

1. Enter the vi editor and create a PCL function in a file called `write_number.pcl`.

There are quick reference pages for vi and UNIX in the appendix of this workbook for your reference.

2. Compile the function.

Start the PCL compiler by typing **p3pclcomp** in your xterm window.

3. Enter the command **!!input write_number.pcl** at the p3pclcomp

prompt:

```
%p3pclcomp
P3/PCL Compiler Release 1.4
Exit or ctrl-d to quit
-> !!input write_number.pcl
Compiling: write_number
Compiled: write_number
->
```

All the error messages and diagnostics will be written to the xterm window

If a function is to be used repeatedly, it is possible to add it to a library. To create a library with this function, or add this function to an existing library, enter this command instead:

```
!!compile write_number.pcl into pat304.plb
```

Then continue to add each new function to the library. When the functions inside a given library are to be used, the library should be referenced using the following command in the PATRAN command window:

```
!!!library pat304.lib
```

Type exit at the p3pclcomp prompt:

```
-> exit
```

4. Start MSC/PATRAN by typing p3 at the xterm prompt

5. Enter the command:

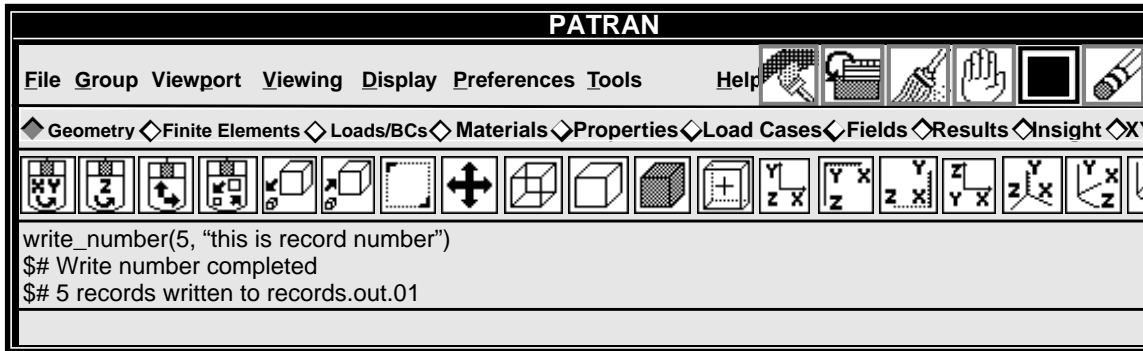
```
!!input write_number.pcl
```

6. Then enter the following at the command line:

```
write_number(5,"this is record number")
```

Once executed, PATRAN should return this message

```
 $#5 records written to records.out.01
```



Change to your xterm window and type:

```
%more records.out.01
```

The file should contain:

```
this is record number 1
this is record number 2
this is record number 3
this is record number 4
this is record number 5
```

Sample Solution:

```
FUNCTION write_number(max_number, leading_text)

/* Purpose: This function opens an external file in the
 * current working directory called records.out and writes
 * max_number records of a format as follows:
 *
 * this is record number 1
 * this is record number 2
 * this is record number 3
 * this is record number 4
 * this is record number 5
 *
 * (if leading_text is "this is record number"
 * and max_number = 5)
 *
 * Input: max_number          I      max_number of records
 * leading_text              S      prefix on each record
 *
 * Output: none
 *
 * Side Effects:
 * The file records.out is either created or the version
 * incremented and the above described records are written to
 * it.
 *
 * Errors:
 * Return 0, no error
 * Otherwise it is a file utility error
 *
 */

    INTEGER max_number
    STRING leading_text[]

    STRING filename[80], outstr[720]
    INTEGER rec, channel

    /*
     * Build a filename with version number and open the
     * external file
     */

    file_build_fname("", "records", "out", "NV", filename)

    text_open(filename, "NOW", 0, 0, channel)

    /*
     * Loop to max_number and write concatenated string
     */

    FOR (rec = 1 TO max_number)
        outstr = leading_text//" "//STR_FROM_INTEGER(rec)
        text_write_string(channel, outstr)
    END FOR

    /*
     * Close the external file
     */

    text_close(channel, "")
```

```
/*
 * Tell user complete
 */

ui_write("Write_number completed.")
ui_write(STR_FROM_INTEGER(rec-1)//" records written to "//@
filename)
RETURN 0

END FUNCTION
```

