IDL Lab #5: The Other IDL Statements

Name:______________________________

IDL Lab #5: A Sub-component of FOR 504 Advanced Topics in Remote Sensing

The objectives of this laboratory exercise are to introduce the student to

- The other IDL Statements, namely:
  - CASE
  - SWITCH
  - BREAK
  - REPEAT
  - CONTINUE
  - GOTO

The tasks provided within this lab are designed to help the student better understand the practical details of programming in IDL and will help you prepare for the class assessment.

If you have problems: ASK!

Location: RS/GIS Lab

Login: XXXX
Password: XXXX

For further reading in this lab exercise, please refer to pages 97-110 (Control Statements) in *Practical IDL Programming*, by Liam Gumley, Academic Press, 2002.
Before you start:

Double click the ENVI icon on the desktop:

This starts both ENVI (*The Environment for Visualizing Images*) and the IDL (*Integrated Development Language*) programming interface

Ignore the ENVI toolbar but don’t close it as this closes IDL as well.

1. The IDL Statements

When you run a computer program, the compiler executes a set of commands in a specific order. In IDL, the _statements_, ‘control’ the order in which the commands are executed by the compiler.

In the earlier labs you have already met a few of the IDL statements, namely the IF statement (lab 2); the FOR loop (lab 3); the DO WHILE loop (lab 4); and the RETURN statement (lab 4).

In this section we will look at the rest of the IDL statements (Table 5.1) and write little programs to demonstrate their use.

A) The CASE Statement

The case statement is a useful programming tool in IDL that can be used to execute a statement or a block of statements depending on a choice that you enter at the command line.

---

### Table 5.1 The different statements and their uses in IDL (Adapted from Gumley (2002)).

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description/Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF</td>
<td>Lab 2: If a statement is true – execute a statement</td>
</tr>
<tr>
<td>FOR</td>
<td>Lab 3: Repeat statements for a specified number of times</td>
</tr>
<tr>
<td>WHILE</td>
<td>Lab 4: Repeat statements while a condition is true</td>
</tr>
<tr>
<td>RETURN</td>
<td>Lab 4: Return a variable from a function or procedure</td>
</tr>
<tr>
<td>CASE</td>
<td>Select an option (case) form a list depending on the input</td>
</tr>
<tr>
<td>SWITCH</td>
<td>Return to a case in a list of cases</td>
</tr>
<tr>
<td>REPEAT</td>
<td>Repeat statements until a condition is true</td>
</tr>
<tr>
<td>GOTO</td>
<td>Go to a certain label</td>
</tr>
<tr>
<td>BREAK</td>
<td>Break out of the loop or statement</td>
</tr>
<tr>
<td>CONTINUE</td>
<td>Continue the program and allow the next iteration of the loop</td>
</tr>
</tbody>
</table>
Example 1: Using a CASE Statement in the circle_stats program

Remember the function based circle_stats program in lab 2. In this version of the program we are going to use a CASE statement to calculate and print to the screen only the area, only the circumference, or both answers.

```plaintext
FUNCTION ARE, dummy
RETURN, (3.14159*dummy*dummy)
END

FUNCTION CIR, dummy
RETURN, (2*3.14159*dummy)
END

PRO circle_stats_case
choice = ""
PRINT, 'Hello World!'
READ, choice, prompt='Do you want the Area (a), the Circumference (c), or Both (b):'
cmdlist = ['a', 'c', 'b']
IF (strlowcase(choice) EQ cmdlist(0)) THEN index = 0
 IF (strlowcase(choice) EQ cmdlist(1)) THEN index = 1
 IF (strlowcase(choice) EQ cmdlist(2)) THEN index = 2
x = 2.0 ; The Radius of the Circle
area = ARE(x)
circum = CIR(x)
CASE index of
  0: PRINT, 'The Area of a Circle with Radius', x, 'is ', area
  1: PRINT, 'The Circumference of a Circle with Radius', x, 'is ', circum
  2: BEGIN
      PRINT, 'The Area of a Circle with Radius', x, 'is ', area
      PRINT, 'The Circumference of a Circle with Radius', x, 'is ', circum
   END
ELSE : PRINT, 'The Valid Display Commands are: ', cmdlist
ENDCASE

END
```

Lets now look at what we wrote in more detail.
Line 15: reads in from the command prompt a ‘string’ variable entered by the user
Line 16: cmdlist contains a ‘list’ of acceptable entries
Lines 27 – 24: execute statements relating to each index number

Another useful application of CASE statements, is to have the different cases triggered by the results of relational and Boolean operators (e.g. 1=TRUE of 0=FALSE):

X = 2

CASE 1 of
  (x LT 2): PRINT, ‘Too Low’
  (x GT 2): BEGIN
    PRINT, ‘To high’
    END
  ELSE: PRINT, ‘Just Right’
ENDCASE

B) THE REPEAT STATEMENT

This statement executes one or multiple statements until a condition is TRUE:

REPEAT statement UNTIL condition

Alternatively you can write the statement as:

REPEAT BEGIN
  statement(s)
ENDREP UNTIL condition

C) THE GOTO STATEMENT

This statement makes the program jump to a specific location in the program – BE CAREFUL most programming languages have the GOTO command but it is the cause of a lot of errors.

D) THE SWITCH STATEMENT

The SWITCH statement is very similar to the case statement and again can be used in menus. However, the difference is in the CASE statement you entered ‘strings’ at the command line, while the SWITCH statement can deal with numbers.

If you enter a number into the program, the SWITCH statement will look for the first matching number and execute the statements associated with that number.

However, unlike the CASE statement, the program does not terminate after it finds a match, but rather the program executes the next statements in the list until either a BREAK or END function
is reached. If no matches are found the statements under ELSE are executed. Examples where the SWITCH statement can be of use included instances where several different numerical inputs each have to do the same statement:

Example 1: Using a SWITCH Statement in a Menu

```idl
PRO switch_test, option

SWITCH option OF

1: begin
  PRINT, 'You chose a number between 1 and 3'
  BREAK
END

4: begin
  PRINT, 'You chose a number between 6 and 7'
  BREAK
END

7: begin
  PRINT, 'You chose a number between 6 and 7'
  BREAK
END

ENDSWITCH

END
```

In this example if you compile the program and then type `switch_test, 1` at the IDL command prompt; then the SWITCH will start at line 5.
- The program will then jump to the next line (line 6) and execute the statements if any (not this time as no begin).
- The program will then jump to the next line (line 7) and execute the statement following the ‘begin’.
- The program will then see the BREAK statement, telling it to end the SWITCH loop
- Next the program ends with the END at line 10.

However, say you entered `switch_test, 6` at the IDL prompt, then the SWITCH will start at line 13, and proceed to the END at line 17.
Entering a value of 5, will cause the program to do leave the SWITCH LOOP and go to line 22.
E) THE BREAK STATEMENT

As you might have guessed from the previous example; the BREAK statement causes the program to exit from a loop, case, or switch statement.

F) THE CONTINUE STATEMENT

The CONTINUE statement causes the NEXT iteration of a loop to be executed. The program then ignores any statements within the current iteration. A useful example of this statement as outlined by Gumley (2002) on page 109 arises when you a reading an IDL source file and you want to ignore any comments:

```
e.g.
WHILE the file has not ended
    Read in the text
    IF (text eq ';') THEN CONTINUE
    Write out the text to a new file
ENDWHILE
```

i.e. is the text is a comment, then the third line in the WHILE loop is skipped.

Task #1 Write a program to compute the days in the month with correct accounting for leap years

[Hint: A leap year is divisible by 4. In the case that the year is also divisible by 100; then to be a leap year it must be divisible by 400. Note if you get really stuck, the answer to this task can be found in Gunley page 107 – but first please try it yourself.]

Task #2 Write a program that reads in the following IDL program code and outputs a new code that has no comments.

```
1   PRO example
2   ;
3   ; A Simple commented Example of Hello World
4   ; Here’s the Print Statement
5   ;
6   PRINT, ‘Hello World!’
7   ; Time to End the Program
8   END
```

[Hint: You will have to use the following functions and expressions: EOF, LUN, strcompress, and strupcase. Check the IDL online help for the syntax. The solution for this task is on page 109 of Gumley (2002).]