Quotes

• Genius is 1 percent inspiration and 99 percent perspiration!
• There is no secret to success—it is a result of preparation, hard work, and learning from the mistakes—Colin Powell
• If you are going to achieve excellence in big things, you develop the habit in little things. Excellence is not an exception, it is a prevailing attitude—Colin Powell

Learning Objectives

• Why decisions?
• IF-THEN-ELSEIF-ELSE conditional operations
• Combining conditions
• Loops
• Examples

Program to Find Real Roots of a Quadratic Equation

Sub Quad()
' A program to compute the real roots of a quadratic equation, ax² + bx + c. These coefficients are prompted as input from the keyboard. The discriminant is calculated but no decision made based on its value. An error will result if the square root of a negative discriminant is attempted. Otherwise the two real roots r1 and r2 will be computed and output to the screen.
Dim a as double
Dim b as double
Dim c as double
Dim d as double
Dim r1 as double
Dim r2 as double
a = Cells(1,2)
b = Cells(2,2)
c = Cells(3,2)
d = b*b – 4*a*c
Cells(5,2)= d
r1 = (-b + Sqr(d))/(2*a)
r2 = (-b – Sqr(d))/(2*a)
Cells(6,2)=r1
Cells(7,2)=r2
End Sub

Why Decisions?

• Need to make decisions in programs
• In the quadratic equation code, 
a*x² + b*x + c = 0
has real roots if the discriminant,
d = b² - 4*a*c
is >= 0
Otherwise complex roots

VBA Comparisons

• VBA comparisons made with If-Then-Else statements
Syntax:
If (conditions) Then
    .......
Else
    .......
Endif

VBA Comparisons

• If the condition is true the lines immediately following If are performed
• If the condition is false control shifts to the lines following Else
• If the operation is only performed if the condition is true (i.e. no alternative) Else and the lines between it and Endif not needed
Example of Real or Complex Roots of Quadratic
\[ d = (b^2 - 4ac) \geq 0 \]
If \( d \geq 0 \) Then
\[
\begin{align*}
    x_1 &= \frac{-b + \sqrt{d}}{2a} \\
    x_2 &= \frac{-b - \sqrt{d}}{2a}
\end{align*}
\]
Else
\[
\begin{align*}
    r &= \frac{-b}{2a} \\
    \text{cpx} &= \frac{\sqrt{|d|}}{2a}
\end{align*}
\]
Endif

Real roots
Complex roots

VB condition operations

- VBA comparison operations are:
  - = equal
  - > greater than
  - < less than
  - >= greater than or equal
  - <= less than or equal
  - <> not equal

Combining Statements

- Operations are combined with and and or
- and means both the conditions have to be true
- or means only one of the conditions has to be true.
- () added to group operations

And/Or Example

- Example: operation performed if either \( d = 0 \) or \( d > 4 \)
- Programming tip: use parentheses to make the conditions clear
  If \((d = 0) \) or \((d > 4)\) Then
    
    Endif
  
- Example: Letter grade B is assigned for students whose final score is greater than 80 and less than 90
  If \((\text{score} < 90)\) and \((\text{score} > 80)\) Then
    grade = "B"
  Endif

IF THEN ELSEIF

- This is used if there are multiple conditions.
- Syntax:
  If (condition1) Then
    
  Elseif (condition2) Then
    
  Else
    
  Endif

IF THEN ELSEIF

- If statement is true the lines immediately following If are performed
- If statement is false control shifts to the lines following Elseif
- There could be several conditions and each condition will begin with an Elseif statement
- If none of the conditions are satisfied, the statements after Else are executed.
Dr. T.P. Clement  
CE 3010 class notes

Ordering two numbers

• Test two numbers and put the large number in the bottom cell (swap logic)
  a = Cells(1, 1)
  b = Cells(2, 1)
  If (a > b) Then 'swap
    Cells(1, 2) = b
    Cells(2, 2) = a
  Else 'Do not swap
    Cells(1, 2) = a
    Cells(2, 2) = b
  End If

Loops

• Many computer calculations performed repeatedly and such repetitions can be facilitated using loops
• Loops can be mixed with conditions to perform powerful calculations

How to use Loops?

• Example:
  • Num1, Num2, and I are integer variables
  For I = Num1 to Num2
    ........
    ........
    Next I
  It is important that you declare I, Num1, and Num2 as integers

Nested Loops

For I = Num1 to Num2
  For J = Num3 to Num4
    ........
    ........
  Next J
Next I

Generate data for drawing a line

• Y = mx + c where x = {0,0.5,1,0.5,1,0,1.5,......,9.5,10}
• We will use an integer step instead of 0.5
  Dim I as Integer
  Dim x as Single, y as Single, m as Single, c as Single
  m = 0.5
  c = 4.2
  For I = 0 To 21
    x = I * 0.5
    y = m * x + c
    Cells(I + 1, 1) = x
    Cells(I + 1, 2) = y
  Next I

How to calculate the sum of 1 to n

• Increment may be negative, i.e., can count down
• Example: want to add the numbers from 1 to 11:
  Dim j as Integer, sum as Single
  Sum = 0# 'must initialize sum
  For j = 1 to 11
    sum = sum + j
  Next j
  Since no step value, j incremented by 1

Sum is an accumulator
Accumulators Explained

• Sum=0
• When j=1
  - sum=0+1
• When j=2
  - Sum=0+1+2
• When j=3
  - Sum=0+1+2+3
• When j = 11
  - Sum=0+1+2+3…….+11

Use of Step (but try to avoid using this option)

• If we want sum only the odd numbers between 1 to n then we can use

  Sum = 0# ' must initialize sum
  For j = 1 to 11 step 2
    sum = sum + j
  Next j

  This would increment j by 2, e.g. j = 1, 3, 5, etc.

  Note, default value of step is equal to 1

Factorial Example

• Compute the factorial of specified integer number using a For-Next loop

  Fact = 1
  'must initialize the product
  n = 4
  For j = 1 to n
    Fact = Fact *j
  Next j

  Note: Fact acts here as another form of accumulator; instead of addition, we multiply all
  the numbers up to n (the definition of factorial)

Loops and IF-then-Else example

DIM I as integer
DIM score as single
DIM grade as string

For I = 1 to 4
  score = cells(I,1)
  if (score > 90) then
    grade = "A"
  elseif (score > 80) then
    grade = "B"
  else
    grade = "F"
  end if
  cells(I, 2) = grade
Next I

Summary of Loops

• Loops may be placed inside other loops
• An inner loop must begin and end inside an outer loop
• Conditions can be placed within loops
• You must indent (using spaces or tabs) all the loops, and if-then-else conditions to clearly mark their start and end locations

For Loops and Spreadsheet Cells

• We can use For loops to output results to different cells
• Example: Write a code to evaluate the factorial of 20 numbers (1 to 20) and output the results to appropriate cells
• Outer loop to control the number for which we are evaluating the factorial
• Inner loop will perform product and evaluate the value of the factorial
• Output the factorials to cell location changing from 1 to n
Multiple Factorials Example

REM Prints factorial of 1 to 20
i, j, xoffset are integers, and Fact single
cells(1,1) = "Number"
cells(1,2) = "Factorial"
xoffset = 1 'Offset used to manage cell location
For i = 1 To 20
fact = 1
For j = 1 To i
fact = fact*j
Next j
Cells(xoffset+i,1) = i
Cells(xoffset+i,2) = fact
Next i

Using Loop index or counter with Cells Statement

• When reading/outputting inside the loops, the loop index (i, j, etc.) could be used to automatically increment the cell row or column numbers.
• The statement Cells(i,1) will read the first number from cells(1,1) for i = 1 and automatically shifts the row to 2 when i=2, and so on...
• If the value should come from a different cell, add an offset to the i value.
• For example: If the first values are in cells(31,10), cells(32,10), cells(33,10). Then define xoffset = 30 then For i=1 to 3 read from cells((xoffset+i),10)
In some cases, like matrices you might need to change both row and column indices.

Bubble sort example

n = 5
For i = (n - 1) To 1 Step -1
For j = 1 To i
If Cells(j, 1) > Cells(j + 1, 1) Then
a = Cells(j, 1)
b = Cells(j + 1, 1)
Cells(j, 1) = b
Cells(j + 1, 1) = a
End If
Next j
*put a break point here and check the values as "i" decreases*
Next i
Try for 24, 28, 22, 27, 21

Results for i = (n-1) to 1

<table>
<thead>
<tr>
<th>i value =&gt;</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>24</td>
<td>22</td>
<td>22</td>
<td>21</td>
</tr>
<tr>
<td>28</td>
<td>22</td>
<td>24</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>22</td>
<td>27</td>
<td>24</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>27</td>
<td>21</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>21</td>
<td>28</td>
<td>28</td>
<td>28</td>
<td>28</td>
</tr>
</tbody>
</table>