▸ Declaration of constants

• Constants are variables which do not change their value during the execution of the program (UDF).

· Constants are used to keep the programming structure clear and to avoid tedious re-typing or time consuming re-calculations.

· You can declare constants

i) such that they are only available inside the program or

ii) such that they are available in the entire worksheet.

Syntax: i) **Const** name [**as** type] = value

ii) **Public Const** name [**as** type] = value

Function .....

It is important to do the **Public Const** statement before the **Function** statement.

· Expl. a) Const Pihalf = 1.570796327

b) Const Errmess as string = "Division by zero!!!"

c) Public Const Errmess as string = "Division by zero!!!"

56

▸ Program Structures

• So far we have only seen *sequential structures* (line by line)

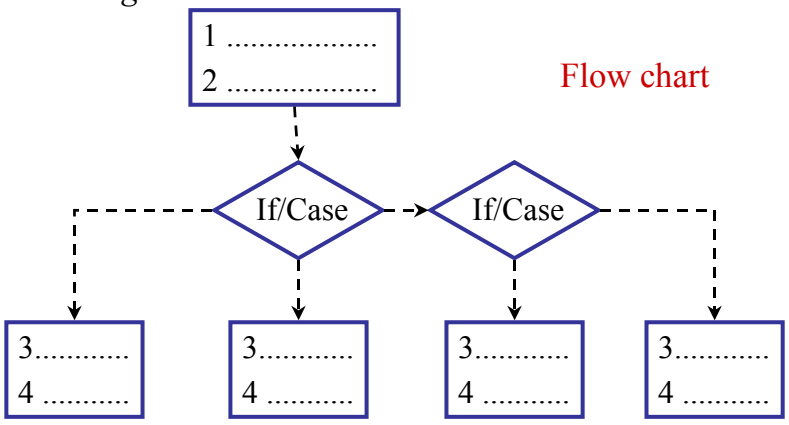
1 .....

2 .....

3 .....

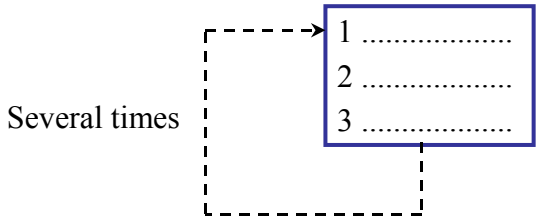
• You can change this way of execution by *control structures*

· *branching* or *decision structures*

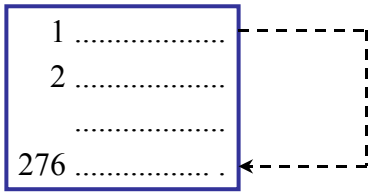


57

· *looping (repetition structures)*



· *controlled GOTO*



• It is useful to draw flow charts in order to keep track of the logic of the program structure. You do not need to write all comments in detail, but it suffices to write general statements in words. 58

▸ The IF-structure

• The IF-structure allows you to change the flow of your program depending on various conditions. The logic of this structure is very similar to the discussed Excel built-in IF-function.

Syntax1: **If** condition **Then**
[statements]
[**ElseIf** condition´ **Then**
[elseifstatements]]... } can be repeated many times
[**Else**
[elsestatements]]
End If

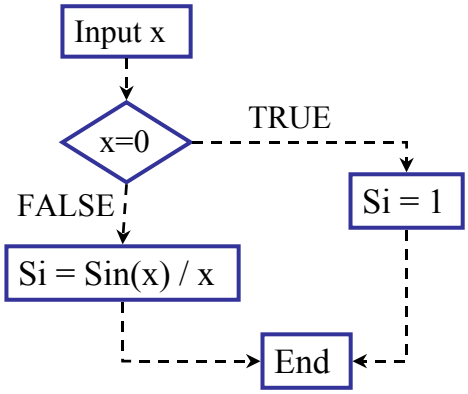
- condition(´) ≡ expressions which are true of false
- statements ≡ valid VBA commands
- elseifstatements ≡ executed when condition´ is true 59
- elsestatements ≡ executed when no previous condition is true

- Examples:

a) Write a UDF which produces the function

$$Si(x) = \begin{cases} \frac{\sin x}{x} & \text{for } x \in \mathbb{R}\backslash 0 \\ 1 & \text{for } x = 0 \end{cases}$$

```
Function Si(x)
    If x = 0 Then
        Si = 1
    Else
        Si = Sin(x) / x
    End If
End Function
```



· Recall from Lab 1 Task 3 that this function also can be produced by using Excel built-in functions as

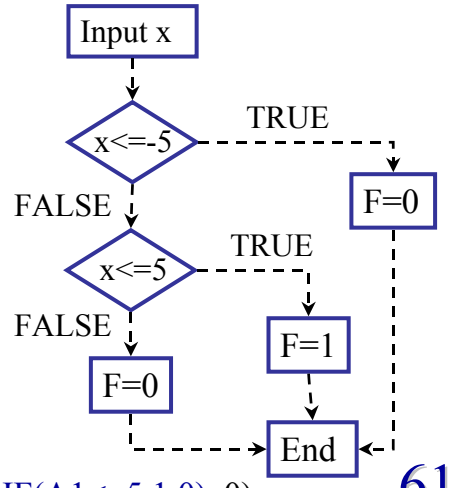
=IF(x=0,1,SIN(x)/x)

60

b) Write a UDF which produces the function

$$F(x) = \begin{cases} 0 & \text{for } x \le -5 \\ 1 & \text{for } -5 < x \le 5 \\ 0 & \text{for } x > 5 \end{cases}$$

```
Function F(x)
    If x <= -5 Then
        F = 0
    ElseIf x <= 5 Then
        F = 1
    Else
        F = 0
    End If
End Function
```



· Recall page 32: =IF(A1>-5, IF(A1<=5,1,0) ,0)

61

c) Write a UDF which determines whether a certain date falls on a weekend or not!

```
Function WE(x As Date) As String
    Dim temp As Integer
    temp = Weekday(x)
    If temp = 1 Or temp = 7 Then
        WE = "That day falls on a weekend."
    Else
        WE = "That day is a weekday."
    End If
End Function
```

· Format the cell A1 as date and enter 01/11/2005 or Now()
· "=WE(A1)" → That day is a weekday.
· Note that we declared all variable types.

62

Syntax2: **If** condition **Then** [statement1] : [statement2]: ...

· Just one line! The VBA statements are carried out when the condition is TRUE. Several statements are separated by ":".
· Expl.: The function F(x) can also be produced by:

```
Function Ftwo(x as single) as integer
    Ftwo = 1
    If x <= -5 Then Ftwo = 0
    If x > 5 Then Ftwo = 0
End Function
```

Syntax3: **IIf** (*condition*, value for true, value for false)

· Same syntax as for built-in functions with IF → IIF
· Expl.: The function in Expl. a) can also be produced by:

```
Function Fthree(x)
    Fthree = IIf(x = 0, 1, Sin(x) / x)
End Function
```

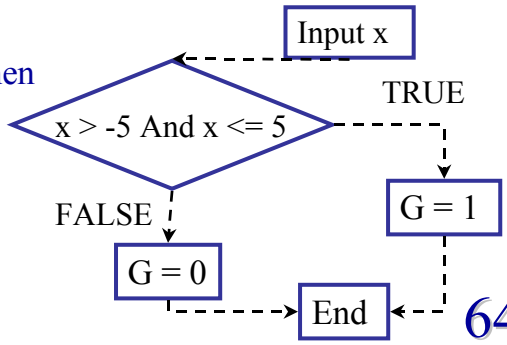
63

- Boolean operators
  - Just as for built-in functions one can use boolean operators to create more complex conditions.

    Syntax: condition1 **And** condition2 **And** condition3

    condition1 **Or** condition2 **Or** condition3
  - The logic is the same as for built-in functions.
  - Expl.: The function F(x) can also be produced by

```
Function G(x)
  If x > -5 And x <= 5 Then
      G = 1
  Else
      G = 0
  End If
End
```

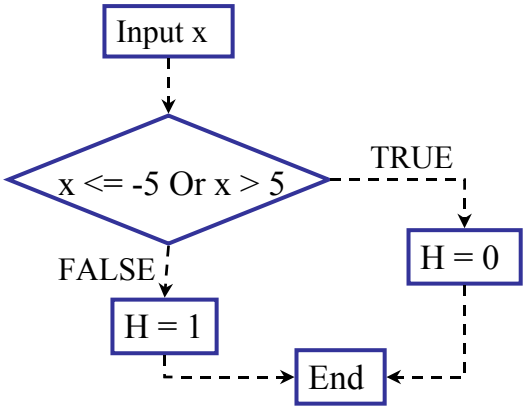


64

  - We need to call it differently when it is on the same WS, e.g. G(x).

- Using "Or" F(x) can be produced by

```
Function H(x)
  If x <= -5 Or x > 5 Then
      H = 0
  Else
      H = 1
  End If
End
```



- You can also use "Not" and produce the same logical structures as with built-in functions, e.g.

```
Function Fnot(x)
    Fnot = IIf(Not (x <= -5 Or x > 5), 1, 0)
End Function
```

65

➤➤➤ The progress test is between:

2nd-13th of January 2006

Find out in time the room where you have to go!

- The test is open book, that means you can take all your notes, Lab-sheets and solutions.
- You will have a computer at your disposal, which you can use to verify and develop your answer.
- You have to write down your answer into an answer booklet which will be provided to you.
- There are two of these tests (exams) each one counting 50% towards your final mark for this course module.
- The questions will be similar in style to the ones of the Lab-sessions.

66