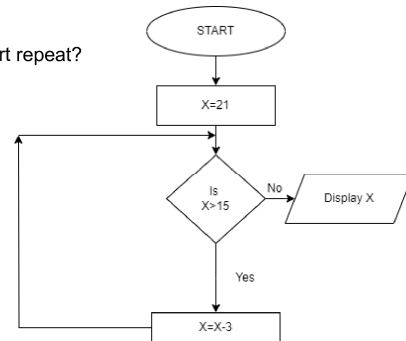


8. How many Times will the following Flowchart repeat?
9. Write down the Pseudocode



10. Write the respective outputs and number of repetitions for the given Pseudocode

a) Begin
A=1
B=5
While A<=B
A=A+1
End while
Display A
End

b) Begin
A=1
C=2
Repeat
C=C+1
A=A+C
Until C>3
Display A
End

11. Draw Flowcharts for the above mentioned Pseudocodes

#ICT_Batman

NAME:

Pascal + Repetition

What is Pascal?

A high level programming language that is imperative and facilitates procedural programming.

What is Repetition?

Repetition is about repeating the steps over and over again until a certain condition is satisfied



076 212 6767

Kevin Noel

BCS (UK), Bachelors of Computing (UG)

Kevin Noel
BCS (UK), Bachelors of Computing (UG)

076 212 6767

Repetition

Repeating the same steps over and over again until an expected condition is satisfied refers to repeating

For an example:

We try to save money targeting a specific amount of money in order to buy a good that we wish to. We keep repeating the process of saving money so that, we achieve the expected amount. As soon as we achieve the amount, we stop repeating. This is because, the condition is satisfied

There are two types of main repetition structures:

- While- do
- Repeat until

While do: Works like a **Debit Card**. It checks the balance before executing the rest of the steps.

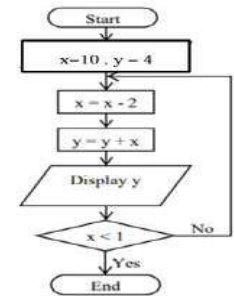
If it is **YES**, Then the repetitive steps *will be executed*

If it is **NO**, The repetition *will be ended*

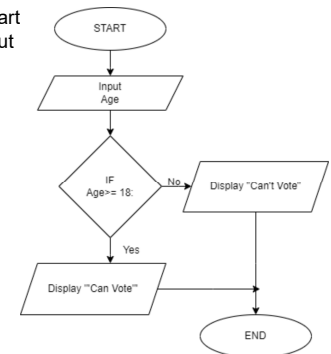
1) Printing/Displaying numbers from 1 to 10

Who cares Programming, unless we do?

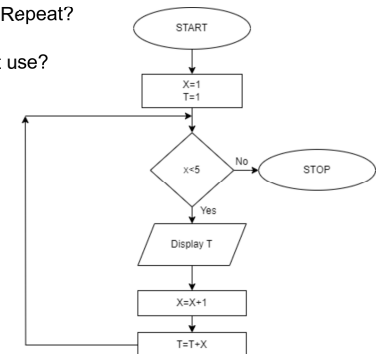
1. Write down the series of Outputs of "y"
2. Write down the Pseudocode for the following Flowchart



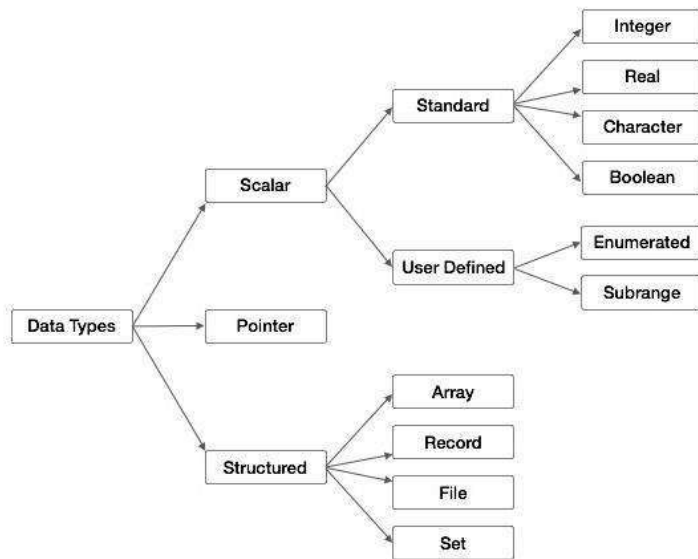
3. Write down the Pseudocode for the following flowchart
4. If the Age of a Person is 18, what would be the output



5. How many Times will the following Flowchart Repeat?
6. Write down the Pseudocode
7. What Repetition structure does this Flowchart use?

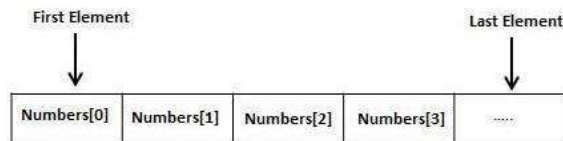


DATA TYPES IN PASCAL



Arrays in Pascal

- Pascal programming language provides a data structure called the array, which can store a fixed-size sequential collection of elements of the same type.
- An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.



Repeat- Until: Works like a **Credit Card**. It executes the steps first and then checks the balance. Repeats the steps until the allocated amount is overridden
If it is **NO**, Then the repetitive steps *will be executed*
If it is **YES**, The repetition *will be ended*

1) Printing/Displaying numbers from 1 to 10

In **exception**, we have,

- **FOR – DO** repetition structure

For- Do: Works to *simplify the coding structure when implementing repetition*

When writing Pseudocodes: FOR-DO Loop can be implemented easily than the other repetition structures

But, it is **mandatory for you to know While-do and Repeat Until to understand “FOR- DO”**

1. Printing/displaying numbers from 1 to 10 – Pseudocode

2. Finding the total of 10 numbers – Pseudo code

3) *Finding the Total of any 10 numbers*

4) *Finding the greatest number from any 5 numbers input*

Writing a Basic Pascal Program

Finding the sum of two numbers

We don't use Input and Display in Pascal instead, we use Write and Read

Input- Read
Output- Write

Lets code Pascal Programs now!

2) *Displaying/Printing numbers from 1 to 10*

3. Finding the total and Average of any 10 numbers

4. Finding the greatest number from any 5 numbers input by user

Pascal *(named after Blaise Pascal)*

- As of now, it has all been about planning on “*what to code*” and “*how to code*” using *Flowcharts/Pseudocodes*

Ex: Flowcharts and Pseudocodes are like developing a house plan, the house is built based on the house plan

“NOW IT’S UP TO BUILDING THE HOUSE” →

“CODING THE PROGRAM BASED ON THE ALGORITHM”

Basic Structure of Pascal

```
program {name of the program}
uses {comma delimited names of libraries you use}
const {global constant declaration block}
var {global variable declaration block}

function {function declarations, if any}
{ local variables }
begin
...
end;

procedure { procedure declarations, if any}
{ local variables }
begin
...
end;

begin { main program block starts}
...
end. { the end of main program block }
```

1. Program Name
2. Uses – Ex: Input, Output
3. Variables used
4. Functions/Program
5. End of the program