Pseudo language Constructs.

(1) **If-then-else construct :** The general form of this construct which is used to provide selection of actions is

```
If condition
then step 1
```

When an instruction using this type of construct is executed, condition determines which of the step 1 and step 2 is to be executed. If condition is true, step 1 is executed, otherwise (i.e., if condition is not true) step 2 is executed.

A particular case of this construct does not have the word else. The general form of this construct is

```
If condition
then step
```

Clearly when this form is used, no action is taken when condition is false, and step is executed when condition is true. In other words the following constructs are equivalent as they do the same thing.

If condition **then** step	and	**If** condition **then** step

(2) **Repeat until construct:** The general form of this construct is

```
Repeat
   Part of the algorithm
```

This construct is used when repetition of certain action is required. Note that "Part of algorithm" is always executed at least once as the condition is tested at the end, unlike the 'while-do' construct where the condition is tested in the beginning.

(3) **While-do-construct:** This construct is an alternative to the 'repeat-until' construct. The general form of this construct, which is also used to provide repetition of instruction is

while	condition
do	T

Where T is a sequence of instructions. When this construct is executed, condition is evaluated first. If the condition is true, the sequence T of instructions is executed and the condition is evaluated again and so on. If the condition is false, execution of T is skipped and the execution of algorithm proceeds with the portion that appears after T. Thus condition is tested again and again till it is false. Every execution of T modifies some variables in the algorithm and eventually after some repetitions, the condition becomes false. This completes the execution of the 'while-do-construct', the execution proceeds to the portion appearing after T.

(4) **For construct:** When we know in advance how many times a part of the algorithm is to be executed we use 'for construct', whose general form is

For identifier = initial value to test value by increment **Do** S.

The word, **For**, **To,By** and **Do** are reserved words for this construct. Initial value gives the starting value that the identifier should take, when the S is executed. The value of identifier is increased by the increment after each execution. The execution of S continues until the value of identifier exceeds the test value.