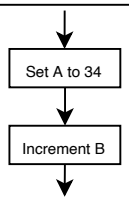
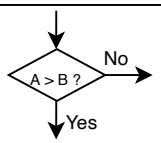
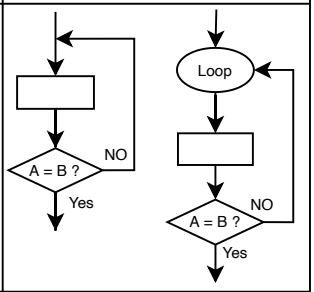
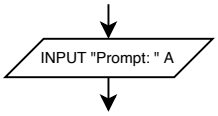
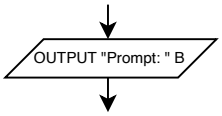
ALevel CS C12 Algorithm design and problem-solving

Structured English: a subset of the English language that consists of command statements used to describe an algorithm
Pseudocode: a way of using keywords and identifiers to describe an algorithm without following the syntax of a particular programming language
FlowChart: shapes linked together to represent the sequential steps of an algorithm

	Structure English	Pseudocode	Flowchart
Assigmen and Sequence	SET A TO 34 INCREMENT B	A ← 34 B ← B + 1	
Selection	IF A IS GREATER THAN B THEN ... ELSE...	IF A > B THEN ... ELSE ... ENDIF	
Repetition	REPEAT UNTIL A IS EQUAL TO B ...	REPEAT ... UNTIL A = B	
Input	INPUT A	INPUT "Prompt: " A	
Output	Output "Message" Output B	Output "Message", B	

Pre-condition loop:

```
SecretNumber ← Random
INPUT Guess
NumberOfGuesses ← 1
WHILE Guess <> SecretNumber DO
   IF Guess > SecretNumber
      THEN
         // the player is given the message to input a smaller number
   ENDIF
   IF Guess < SecretNumber
      THEN
         // the player is given the message to input a larger number
   ENDIF
   INPUT Guess
   NumberOfGuesses ← NumberOfGuesses + 1
ENDWHILE
OUTPUT NumberOfGuesses
```

Post-condition loop:

```
SecretNumber ← Random
NumberOfGuesses ← 0
REPEAT
   INPUT Guess
   NumberOfGuesses ← NumberOfGuesses + 1
   IF Guess > SecretNumber
      THEN
         // the player is given the message to input a smaller number
   ENDIF
   IF Guess < SecretNumber
      THEN
         // the player is given the message to input a larger number
   ENDIF
UNTIL Guess = SecretNumber
OUTPUT NumberOfGuesses
```

Logic statement:

```
IF Guess = SecretNumber
   THEN
      OUTPUT "Well done. You have guessed the secret number"
   ELSE
      IF Guess > SecretNumber AND NumberofGuesses = 10
         THEN
            OUTPUT "You still have not guessed the secret number"
         ELSE
            IF Guess > SecretNumber
               THEN
                  OUTPUT "secret number is smaller"
               ELSE
                  OUTPUT "secret number is greater"
            ENDIF
      ENDIF
ENDIF
```

REPEAT UNITL:

```
INPUT BiggestSoFar
Counter ← 1
REPEAT
   INPUT NextNumber
   Counter ← Counter + 1
   IF NextNumber > BiggestSoFar
      THEN
         BiggestSoFar ← NextNumber
   ENDIF
UNTIL Counter = 10
OUTPUT BiggestSoFar
```

For Next:

```
INPUT BiggestSoFar
FOR Counter ← 2 TO 10
   INPUT NextNumber
   IF NextNumber > BiggestSoFar
      THEN
         BiggestSoFar ← NextNumber
   ENDIF
NEXT Counter
OUTPUT BiggestSoFar
```

While loop:

```
INPUT NextNumber
BiggestSoFar ← NextNumber
WHILE NextNumber <> 0 DO // sequence terminator not encountered
   INPUT NextNumber
   IF NextNumber > BiggestSoFar
      THEN
         BiggestSoFar ← NextNumber
   ENDIF
ENDWHILE
OUTPUT BiggestSoFar
```

Procedure and Function

```
CALL SetValues

REPEAT
   CALL OutputSpaces
   CALL OutputSymbols
   NumberOfSpaces ← AdjustedNumberOfSpaces
   NumberOfSymbols ← AdjustedNumbeOfSymbols
UNTIL NumberOfSymbols > MaxNumberOfSymbols

PROCEDURE SetValues
   INPUT Symbol
   MaxNumberOfSymbols ← ValidatedMaxNumberOfSymbols
   NumberOfSpaces ← (MaxNumberOfSymbols - 1) / 2
   NumberOfSymbols ← 1
ENDPROCEDURE

FUNCTION ValidatedMaxNumberOfSymbols RETURNS INTEGER
   REPEAT
      INPUT MaxNumberOfSymbols
   UNTIL MaxNumberOfSymbols MOD 2 = 1
   RETURN MaxNumberOfSymbols
ENDFUNCTION

PROCEDURE OutputSpaces
   FOR Count1 ← 1 TO NumberOfSpaces
      OUTPUT Space // without moving to next line
   NEXT Count1
ENDPROCEDURE

PROCEDURE OutputSymbols
   FOR Count2 ← 1 TO NumberOfSymbols
      OUTPUT Symbol // without moving to next line
   NEXT Count2
   OUTPUT Newline // move to the next line
ENDPROCEDURE

FUNCTION AdjustedNumberOfSpaces RETURNS INTEGER
   NumberOfSpaces ← NumberOfSpaces – 1
   RETURN NumberOfSpaces
ENDFUNCTION

FUNCTION AdjustedNumberOfSymbols RETURNS INTEGER
   NumberOfSymbols ← NumberOfSymbols + 2
   RETURN NumberOfSymbols
ENDFUNCTION
```