# AQA Computer Science A-Level
## 4.2.3 Stacks
Past Paper Mark Scheme

| 5 | (d) | | |
|---|---|---|---|
| | | ```
If StackArray is full
   Then Stack Full Error
   Else
       Increment TopOfStackPointer
       StackArray [TopOfStackPointer] ←
             ANumber
EndIf
```

**1 mark** for appropriate `If` structure including condition (does not need both `Then` and `Else`) – Do not award this mark if `ANumber` is put into `StackArray` outside the `If`.
**1 mark** for reporting error in correct place
**1 mark\*** for incrementing `TopOfStackPointer`
**1 mark\*** for storing value in `ANumber` into correct position in array | |

| | | | |
|---|---|---|---|
| | | \* = if the store instruction is given before the increment instruction OR the `If` structure then award **MAX 1** of these two marks UNLESS the item is inserted at position `TopOfStackPointer+1` so the code would work.

**I** initialisation of `TopOfStackPointer` to 0

**A** `TopOfStackPointer=20` / `>=20` for Stack is full
**A** Logic of If structure reversed i.e. If stack is not full / `TopOfStackPointer<20` / `<>20` / `!=20` and `Then`, `Else` swapped
**A** Any type of brackets or reasonable notation for the array index
**DPT** If candidate has used a different name any variable then do not award first mark but award subsequent marks as if correct name used.

Refer answers where candidate has used a loop to find position to insert item into stack to team leaders. | **4** |

# June 2017 Paper 1 Mark Scheme

| 06 | 1 | **Marks are for AO2 (analyse)** | 5 |
|---|---|---|---|
| | | 1. Stack / data structure is used to store the (user's) actions; **A.** by implication | |
| | | 2. Each time an action is completed it is pushed/added onto the <u>top</u> of the stack; | |
| | | 3. unless it is an undo (or repeat) action; | |
| | | 4. When repeat action is used the top item from the stack is used to indicate the action to complete // when repeat action is used the result of peek function is used to indicate the action to complete; **R.** implication that top item of stack is popped/deleted from stack – unless it is clear it is subsequently pushed/added back to the stack  **A.** when repeat action is used a copy of the top item from the stack is pushed/added to the top of the stack | |
| | | 5. When undo action is used the top item is popped/removed from the stack of actions; | |
| 06 | 2 | **Mark is for AO1 (understanding)** | 1 |
| | | Stack empty (error) // (stack) underflow; | |