# 1997 Paper 12 Question 3

## Processor Architecture

The ARM processor allows every instruction to be conditionally executed whereas many processors allow only branches to be conditional. In ARM assembler conditional execution is indicated by one of the following postfix mnemonics:

| Condition code mnemonic | Meaning |
|---|---|
| `EQ` | equal |
| `NE` | not equal |
| `CS` | unsigned higher or same |
| `CC` | unsigned lower |
| `MI` | negative |
| `PL` | positive or zero |
| `VS` | overflow |
| `VC` | no overflow |
| `HI` | unsigned higher |
| `LS` | unsigned lower or same |
| `GE` | greater or equal |
| `LT` | less than |
| `GT` | greater than |
| `LE` | less than or equal |
| `AL` | always execute (default) |

($a$) Using the following C-code excerpt as a basis for argument, briefly explain how conditional execution of every instruction can reduce the size of the code when compared with an instruction set which allows only conditional branches.

```
if (x==0) a=y; else a=y*x;
```

[7 marks]

($b$) What effect do branch instructions have on a processor pipeline which does not perform branch prediction (i.e. as on the ARM 7)? [7 marks]

($c$) What effect do conditional instructions have on the pipeline if the condition fails, and why is this effect preferable to that of short conditional branches (assuming no branch prediction)? [6 marks]