25) a) i)

| A	B	C	Not A. AND B	A OR B	S
0	0	0	1	0	0
0	1	0	1	1	1
1	0	0	1	1	1
1	1	1	0	1	0

ii)
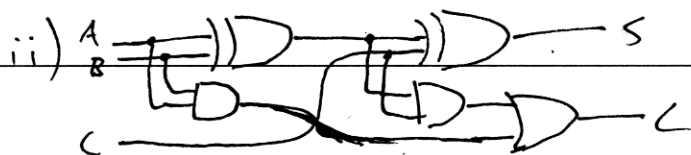
Two half adders can be linked like above to form a full adder.

b) Integer representation can only use whole numbers whether negative or positive. An example of this would be storing the number of days in a month. Floating point representation deals with decimal points and is ~~below~~ better for more precise calculations, such as storing the temperature for each day or for scientific calculations etc.
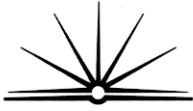
c)i) X: 010110010l   32168421   = right 50mm

  Y: 0110100111   8472168421   = up 83mm

The car moves diagonally up to the right, going

83 mm up and 50mm right.

ii)   10110010₁

      11010011 ⁺
      _____
      ~~10000~~ .

      10000101

                              1001
                        _____
                  1101) 1000101
                         1101
                         _____
                         010001
                           1101
                           _____
                           0100

∴ remainder will be 0100

iii) BEGIN

CHECKLENGTH   CHEEKSOM

Set index = 1

Read Stringln (index)

IF Stringln (index) = 0   THEN

~~WHILE index 2~~

Index = Index + 1

~~Loose1~~

IF   Stringln (index) = 1   THEN

Xdirection = right

Else

X direction = left

ENDIF

Set count = 1

WHILE count < 8

Xdistance (count) ~~index~~ = String (index +1)

Increment index

Increment count

ENDWHILE

ENDIF

Index = Index + 2
~~Increment Index~~

IF index = 0   THEN

Increment index

IF String (index) = 1 THEN

    Ydirection = up

Else

    Ydirection = down

ENDIF

Set count = 1

WHILE count < 8

    Y distance (count, ~~distance~~) = String (Index + 1)

    Increment index

    Increment count

    ENDWHILE
ENDIF
    Move Car (X direction, X distance, Y direction, Y distance)

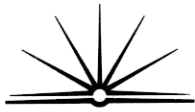~~END~~

~~&~~ END

BEGIN · CHECKLENGTH
Length = 0
Index = 1

WHILE ~~not end of~~ ~~&~~ not end of String

    Increment length
    Increment Index
ENDWHILE

```
IF length <> 30 THEN

    Print "incorrect length"

END IF

END


BEGIN CHECKSUM
    count = 1
    Index = 2
    WHILE index < 9
    ~~Pack~~

Pack1 (count) = StringIn (index)

    increment index

    Increment count

ENDWHILE

Index = 11

Count = 1

WHILE count < 8

    Pack2 (count) = StringIn (index)

    Increment index

    Increment count

ENDWHILE

~~Pack1 / P~~

Remainder = Mod (Pack1 / Pack2)
```

IF remainder <> ~~Checksum~~ String (~~Text~~ 25-29) THEN

~~Check~~ Print "Checksum failed"

END IF

END