q23

a) COTS package works, has been extensively tested (commercial product) and is well documented. A COTS package has lots of help available for the end user as such to develop their product. If a custom-designed solution is produced, the solution may contain errors, be hard to user, and no support will be provided. On the other hand, a custom-designed package may be needed since the COTS package may not be able to do what the company requires. Therefore a customised solution is needed.

b)

i) Internet - The availability and connectability of the Internet has provided a gateway for millions of users to internet. Faster internet connections (such as broadband) has ensured that more tasks can be completed for efficienlty and effectivly. The production of new operating environments

such as Windows XP has enabled users with minimal knowledge to interact easier within a computing environment. Also more features have been provided for expert users. The development of friendlier programing languages with Graphical user interfaces has ensured that more users can program in a programing language an example BASIC to VISUAL BASIC or C to Visual C++.

ii)

Management of the current code. Since anyone from around the world could contribute, the collaberation of all the code into the final program could be hectic. There could be thousands of contributions, and how would each piece of code be connected with the rest of the program. Also, the type of code must be categorised otherwise

code may be forgotten or lost or even left out. Management would also have to keep track of the contributors so that their efforts are recognised in the final product.

c)

i) Reg n

ii)

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	A
10	B
	C

$A1_8 = 81_{10}$

81
16 A 1

Mem 6 = Mem 5 + Mem 7

= 30 + F8

This is because of the lines of code executed in the program defined in the question

STORE (Reg 3, mem 6)

∴ ~~Mem 6 = Reg 3~~

~~Reg 3 = Mem 6~~

Mem 6 = Reg 3

iii) Reg 3 = Mem 5 + Mem 6

= 30 + F8

= 91 + 141

= 232_{10}

I arrived by this by converting hexadecimal into binary

iv) ~~MULTIPLY (Reg n, Reg n, 3)~~

LOAD (Reg 1, Mem 5)

LOAD (Reg 2, Mem 5)

ADD (Reg 3, Reg 1, Reg 2)

STORE (Reg 3, Mem 7)

LOAD (Reg 1, Mem 7)

LOAD (Reg 2, mem 5)

ADD (Reg 3, Reg 1, Reg 2)

STORE (Reg 3, Mem 7)

STOP