1.0)

1.0.1) variables i & j are repeatedly used.
Also, for each "i" iteration, ~~A[i][1]~~ A[i][1] is used for 8000 times
→ A[i][1] is a temporal locality

1.0.2

~~A[i][1] is exhibit exhibit spatial locality because~~
Since the elements are executed / stored in "Row Major Format",
B[i,j] exhibit spatial locality because B[i][j] and B[i][j+1] are ~~relatively~~ "close".
A[j][i] does not exhibit spatial locality because A[j][i] and A[j+1][i] are not "close" because there is 8000 elements (j iteration) between those

2.0)

2.0.1) Cache capacity, C = (Number of Blocks) × (Block size)
= (Number of set) × (Number of ways) × (Block Size)
= $\boxed{S \cdot N \cdot b \text{ (words)}}$
= $\boxed{4 S \cdot N \cdot b \text{ (bytes)}}$ (one word = 4 bytes)

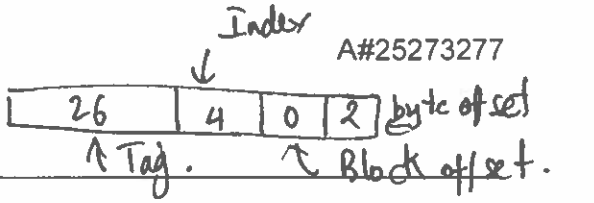
2.0.2).

- In full associative, whole block in cache is considered as one set.
→ S = 1. Hence, word in the main memory can be saved in any block

- $C = S \cdot N \cdot b = Nb$ (S = 1)
→ $\boxed{N = C/b}$

3.0)

3.0.1)



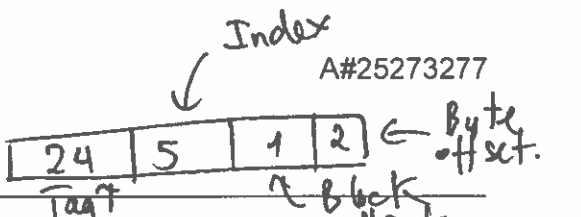
3.0.1) Block size = 1 word
Blocks = 16 Blocks.

Byte offset = $\log_2$(# bytes in a word) = $\log_2(4)$ = 2 bit
Block offset = $\log_2$(# words in a block) = $\log_2(1)$ = 0 bit
Index = $\log_2$(# Blocks) = $\log_2(16)$ = 4 bit
Tag bit = 32 − (2+0+4) = 26 bits.

a 32-bit address w/ value but write version as short

Address (Hex/Binary)	Index	Tag	Data	Hit / Miss
0x74 / ... 0000 0111 0100	1101	000001	M[0x74]	Miss.
0xA0 / 0000 1010 0000	1000	000010	M[0xA0]	Miss
0x78 / 0000 0111 1000	1110	000001	M[0x78]	Miss
0x38C / 0011 1000 1100	0011	001110	M[0x38C]	Miss
0xAC / 0000 1010 1100	1011	000010	M[0xAC]	Miss
0x84 / 0000 1000 0100	0001	000010	M[0x84]	Miss.
0x88 / 0000 1000 1000	0010	000010	M[0x88]	Miss
0x8C / 0000 1000 1100	0011	000010	M[0x8C]	Miss
0x7C / 0000 0111 1100	1111	000001	M[0x7C]	Miss
0x34 / 0000 0011 0100	1101	000000	M[0x34]	Miss
0x38 / 0000 0011 1000	1110	000000	M[0x38]	Miss
0x13C / 0001 0011 1100	1111	000100	M[0x13C]	Miss
0x388 / 0011 1000 1000	0010	001110	M[0x388]	Miss
0x18C / 0001 1000 1100	0011	000110	M[0x18C]	Miss

Tag Index Byte offset



3.0.2

Block Size = 2 words
Blocks = 32 blocks

Byte offset = $\log_2$(# bytes in a word) = $\log_2(4)$ = 2 bits
Block offset = $\log_2$(# words in a block) = $\log_2(2)$ = 1 bit
Index = $\log_2$(# Blocks) = $\log_2(32)$ = 5 bits
Tag = 32 − (2+1+5) = 24 bits

Address (Hex / Bin)	Index	Tag (24 MSB)	Data (Hex)	Hit / Miss
0x74 / ... 0000 0111 0100	01110	...0000	M[70...77]	Miss
0x A0 / 0000 1010 0000	10100	...000 0	M[A0...A7]	Miss
0x 78 / 0000 0111 1000	01111	0000	M[78...7F]	Miss
0x38C / 0011 1000 1100	10001	..0011	M[388...38F]	Miss
0x AC / 0000 1010 1100	10101	..0000	M[A8...AF]	Miss
0x 84 / 0000 1000 0100	10000	0000	M[80...87]	Miss
0x 88 / 0000 1000 1000	10001	0000	M[88...8F]	Miss
0x 8C / 0000 1000 1100	10001	0000	M[88...8F]	Hit ✓
0x 7C / 0000 0111 1100	01111	0000	M[78...7F]	Hit ✓
0x 34 / 0000 0011 0100	00110	0000	M[30...37]	Miss
0x 38 / 0000 0011 1000	00111	0000	M[38...3F]	Miss
0x13C / 0001 0011 1100	00111	0001	M[138...13F]	Miss
0x38B / 0011 1000 1000	10001	0011	M[388...38F]	Miss
0x18C / 0001 1000 1100	10001	0001	M[188...18F]	Miss

Tag

Index Block offset + Byte offset

4.0.1, $t_{cache} = 1\text{cycle} / 2.5\text{GHz} = 0.4\text{ns}$

~~4.0.1~~.

AMAT, Average memory access time $= 0.4 + 0.07 \times 45\text{ns} =$ ~~4.45(ns)~~ **3.55 ns**

4.0.2

~~The average CPI for benchmark =~~

Non-ideal memory system:

load instruction requires 4ns for memory access and 4ns for load

→ 8 ns for load

$CPI_{load} = (4\text{ cycle}_{mem} + 4\text{ cycle}_{load}) = 8$ cycle

$CPI_{store} = (4\text{ cycle}_{mem} + 3\text{ cycle}_{store}) = 7$ cycles

$CPI_{branch} = 3$ cycles.

$CPI_{data} = 4$ cycles.

→ Average CPI for benchmark $= (0.25 \times 8) + (0.15 \times 7) + (0.1 \times 3) + (0.5 \times 4)$

$= 5.35$ ns

4.0.3

Average $CPI_{benchmark} = 5.35\text{ns} + (0.03 \times 45\text{ns}) =$ **6.7ns**