Problem 1 (25 points) In this exercise we look at memory locality properties of matrix computation. The following code is written in C, where elements within the same row are stored contiguously. Assume each word is a 64-bit integer. Answer the following questions based on **Section 5.1** of the textbook.

```
for (j=0; j<100; j++)
    for (i=0; i<300; i++)
        A[j][i] = B[i][0] + C[i][j];
```

(a) How many 64-bit integers can be stored in a 16-byte cache block?

(b) Which variable references exhibit temporal locality?

(c) Which variable references exhibit spatial locality? Locality is affected by both the reference order and data layout.

Problem 1 (25)- **In this exercise we look at memory locality properties of matrix computation. The following code is written in C, where elements within the same row are stored contiguously. Assume each word is a 64-bit integer. Answer the following questions based on Section 5.1 of the textbook.**

```
for (j=0; j<100; j++)
    for (i=0; i<300; i++)
        A[j][i] = B[i][0] + C[i][j];
```

(a) How many 64-bit integers can be stored in a 16-byte cache block?

16 * 8 bits = 128 bits

64-bit integer=cache size/integer bits=128/64 **= 2...(2 64-bit integers per 16-byte block)**

(b) Which variable references exhibit temporal locality?

Variables i, j exhibit temporal locality

(Array B[i][0] may exhibit locality sometimes? because it references the same location with 0)

"Pb1.b(-3) B[i][0] does not exhibit temporal locality.

(c) Which variable references exhibit spatial locality? Locality is affected by both the reference order and data layout.

A[j][i] exhibits spacial locality when referencing because access A[0][0], A[0][1], A[0][2]....is local)

Problem 2 (35 points) Below is a list of 8-bit memory address references, given as **word** addresses. Answer the following questions based on **Section 5.3** of the textbook.

```
0x43, 0xc4, 0x2b, 0x42, 0xc5, 0x28,
0xbe, 0x05, 0x92, 0x2a, 0xba, 0xbd
```

(a) For each of these references, identify the binary word address, the tag, and the index given a direct-mapped cache with 16 one-word blocks. Also list whether each reference is a hit or a miss, assuming the cache is initially empty.

(b) For each of these references, identify the binary word address, the tag, the index, and the offset given a direct-mapped cache with two-word blocks and a total size of eight blocks. Also list if each reference is a hit or a miss, assuming the cache is initially empty.

(c) You are asked to optimize a cache design for the given references. There are three direct-mapped cache designs possible, all with a total of 16 words of data. Which one is better for the above sequence of accesses? Justify your answer.

1. C1 has 1-word blocks,
2. C2 has 2-word blocks, and
3. C3 has 4-word blocks.

Consider **Section 5.4** in addition to 5.3 for this question.

Problem 2 (35)- Below is a list of 8-bit memory address references, given as word addresses.

Answer the following questions based on Section 5.3 of the textbook. `0x43, 0xc4, 0x2b, 0x42, 0xc5, 0x28, 0xbe, 0x05, 0x92, 0x2a, 0xba, 0xbd`

(a) For each of these references, identify the binary word address, the tag, and the index given a direct-mapped cache with 16 one-word blocks. Also list whether each reference is a hit or a miss, assuming the cache is initially empty.

Address	Binary	Tag	Index	Hit/Miss?
0x43	0100 0011	0100	0011	M
0xc4	1100 0100	1100	0100	M
0x2b	0010 1011	0010	1011	M
0x42	0100 0010	0100	0010	M
0xc5	1100 0101	1100	0101	M
0x28	0010 1000	0010	1000	M
0xbe	1011 1110	1011	1110	M
0x05	0000 0101	0000	0101	M
0x92	1001 0010	1001	0010	M
0x2a	0010 1010	0010	1010	M
0xba	1011 1010	1011	1010	M
0xbd	1011 1101	1011	1101	M

Address	Binary	Tag	Index	Offset	Hit/Miss?
0x43	0100 0011	0100	001	1	M
0xc4	1100 0100	1100	010	0	M
0x2b	0010 1011	0010	101	1	M
0x42	0100 0010	0100	001	0	**H**
0xc5	1100 0101	1100	010	1	**H**
0x28	0010 1000	0010	100	0	M
0xbe	1011 1110	1011	111	0	M
0x05	0000 0101	0000	010	1	M
0x92	1001 0010	1001	001	0	M
0x2a	0010 1010	0010	101	0	**H**
0xba	1011 1010	1011	101	0	M
0xbd	1011 1101	1011	110	1	M

Of the C1 and C2 above the best is the one with the lowest miss rate…

C1 misses all therefor has a miss rate of 100%

C2 misses 9/12 which is a miss rate of 75% being the best for the above

Pb2.c(-10)Lack of insertion example for C3.
(-2)Missing the comparison for C3, the result is incorrect.

Problem 3 (15 points) Cache block size (B) can affect both miss rate and miss latency. Assuming a machine with a base CPI of 1, and an average of 1.35 references (both instruction and data) per instruction, find the block size that minimizes the total miss latency given the following miss rates for various block sizes.

Block Size	8	16	32	64	128
Miss Rate	5%	3%	2%	1.5%	1.1%

Answer the following questions based on **Section 5.3** of the textbook.

(a) What is the optimal block size for a miss penalty of 30 × B cycles?

(b) If miss latency was constant (independent of B), what is the optimal block size?

Problem 3 (15)- **Cache block size (B) can affect both miss rate and miss latency. Assuming a machine with a base CPI of 1, and an average of 1.35 references (both instruction and data) per instruction, find the block size that minimizes the total miss latency given the following miss rates for various block sizes. Answer the following questions based on Section 5.3 of the textbook.**

Block Size	8	16	32	64	128
Miss Rate	5%	3%	2%	1.5%	1.1%

(a) What is the optimal block size for a miss penalty of 30 × B cycles?

1+(.05 x 240)= *13*; 1+ (.03 x 480)= *15.4*; 1+ (.02 x 960)= *20.2*; 1+ (.015 x 1920)= *29.8*; 1+ (0.011 x 3840)= *43.24*

lowest latency = optimal.....

block size 8-byte is optimal

(b) If miss latency was constant (independent of B), what is the optimal block size?

if independent of B..... the optimal block size would be the **128-bytes** with the lowest miss rate percentage

Problem 4 (25 points) Consider a **byte addressing** architecture with 64-bit memory addresses.

(a) Which bits of the address would be used in the tag, index and offset in a direct-mapped cache with 512 1-word blocks.

(b) Which bits of the address would be used in the tag, index and offset in a direct-mapped cache with 64 8-word blocks.

(c) What is the ratio of bits used for storing data to total bits stored in the cache in each of the above cases?

(d) Which bits of the address would be used in the tag, index and offset in a two-way set associative cache with 1-word blocks and a total capacity of 512 words.

Problem 4 (25)- Consider a byte addressing architecture with 64-bit memory addresses.

(a) Which bits of the address would be used in the tag, index and offset in a direct-mapped cache with 512 1-word blocks.

offset = $\log_2(64)$ = 6 bits

index = $\log_2(512)$ = 9 bits

tag = 64 - 6 - 9 = 49 bits

Pb4.ab(+2) only correct for index, offset includes byte offset and block offset, missing the orders for each element.

(b) Which bits of the address would be used in the tag, index and offset in a direct-mapped cache with 64 8-word blocks.

offset = $\log_2(64)$ = 6 bits

index = $\log_2(64)$ = 6 bits

tag = 64 - 6 - 6 = 52 bits

c. (-10) blank submission, the problem requires the understanding of the components of the address, the answer should be 32*n bits for storing data, 32*n+53+1 total bits in a row of cache, n is the num of word blocks.

(c) What is the ratio of bits used for storing data to total bits stored in the cache in each of the above cases?

(d) Which bits of the address would be used in the tag, index and offset in a two-way set associative cache with 1-word blocks and a total capacity of 512 words.

offset = $\log_2(64)$ = 6 bits

index = $\log_2(512)$ = 9 bits

tag = 64 - 6 - 9 = 49 bits

d. (-5) wrong for all the parts, missing the orders for each element."