Homework 3

Problem 1

The following table gives the parameters for a number of different caches. Fill in the missing fields in the table for each caches. Recall that **m** is the number of physical address bits, **C** is the cache size (number of data bytes). **B** is the block size in bytes, **E** is the associativity, **S** is the number of cache sets, **t** is the number of tag bits, **s** is the number of set index bits, and **b** is the number of block offset bits.

Cache	m	C	B	E	S	t	s	b
1	32	1024	4	4				
2	32	2048			128	23	7	2
3	32	1024	2	8	64			1
4	32	1024	8	128				
5	32	1024		2	16	23	4	
6	32	1024	32	4				

Problem 2

Consider a 32-bit machine with a 4-way set associative cache. There are 16 sets.
Each block is 32 bytes. Answer the following questions:

1. please fill the following blanks

Cache size: [1] bytes

Field	Length(bit)
Total	32
Tag	[2]
Set	[3]
Offset	[4]

2. Assume the cache is initially empty. There are several memory accesses in order.
Fill the second table and compute the miss rate.

Order	Address	Set	Miss or not(Yes/No)
1	0xe166	[1]	[2]
2	0xe17e	[3]	[4]
3	0xe15a	[5]	[6]
4	0x9d60	[7]	[8]
5	0x9d5f	[9]	[10]

6	0xe142	[11]	[12]

3. Assume the following code is executed on this machine. You should also assume the following:

(1)sizeof(int) = 4

(2)the starting address of array a is 0x100

(3)variable i, j, t are in registers

```
#define M 4
#define N 8
int a[M][N];
int i, j;
for (i = 0; i < M - 1; i++) {
        int t = a[i + 1][0];
        for (j = 0; j < N; j++)
                a[i][j] = a[i][j] * t;
}
```

1) Total number of memory accesses: [1]

2) Miss rate: [2]

3) Suppose you have the chance to double the cache size. Which following scheme is best choice for above code and why? (5')

Scheme A: double the number of sets. (i.e., 32 sets)

Scheme B: double the block size (i.e., 64 bytes)

Scheme C: double the number of ways (i.e., 8-way)