a. Calculate the AMAT for a system with the following properties:
   - L1 cache hits in 1 cycle with local hit rate 20%
   - L2 cache hits in 10 cycles with local hit rate 80%
   - L3 cache hits in 100 cycles with local hit rate 90%
   - Main memory always hits in 1000 cycles

   \[ \text{AMAT} = 1 + (1 - 0.2)(10 + (1 - 0.8)(100 + (1 - 0.9)(1000))) = 41 \]

b. How slow can you go? Your system consists of the following:
   - L1 cache hits in 2 cycles with a miss rate of 20%
   - L2 cache hits in 10 cycles
   - Main memory always hits in 300 cycles

   You want your AMAT to be \(<= 22\) cycles. What does your local L2 miss rate need to be? What is the equivalent global miss rate?

   \[ \text{AMAT} = \text{L1 Hit time} + \text{L1 Miss rate} \times (\text{L2 Hit time} + \text{L2 Local Miss rate} \times \text{L2 Miss penalty}) \]
   \[ 22 \geq 2 + 0.2 \times (10 + X \times 300) \]
   \[ X = 0.3, \text{ or 30}\% \text{ local miss rate} \]
   \[ \text{Global miss rate} = 30\% \times 20\% = 6\% \]
Question 2 (SP15 Final):

a) What shape do the following trade-off curves have? Select a shape and enter its number into the box for each of the graphs. Unless they are the parameters being varied, assume that associativity, capacity and block size are constant. You should assume that the axes are linear.

b) Consider a system with inclusive L1 and L2 caches with 4B cache block size. Assume we have 1 MiB of on-chip memory available and want to determine how much of this memory we should give to the L1 cache and how much to the L2 cache. We will try to minimize the AMAT to do so.

Assume both caches are fully associative with LRU replacement. Their combined capacity is 1MiB (excluding tags and meta-data). You can consider all miss rates approximate.

Say you are running the following program starting from cold L1 and L2 caches:

```c
#define ARRAY_SIZE 256*1024
int a[ARRAY_SIZE];
ton sum = 0; // assume sum, i, and j are stored in registers

for (int i = 0; i < 100000; i++) {
    for (int j = 0; j < ARRAY_SIZE; j++) sum += a[j];
    for (int j = ARRAY_SIZE-1; j >= 0; j--) sum += a[j];
}
```

1) How would we compute AMAT if we had the local L1 miss rate (“L1Miss”), the local L2 miss rate (“L2Miss”) and the memory access time (“Memory”)? Use “H1” and “H2” to represent the L1 and L2 hit times respectively. (We will compute these quantities later in the question)

AMAT = H1 + L1Miss * (H2 + L2Miss * Memory)
2) For the program above, express the local miss rate for the L1 cache in general terms as a function of the L1 cache size (write L1 for the size of L1 in bytes). Hint: The miss rate is 0 for a 1 MiB cache, 0.5 for a 0.5 MiB cache and 1 for a 0 MiB (i.e., no) L1 cache.

LocalMiss1 = 1 – (L1/1MiB)

3) What is the global miss rate for the L2 cache as a function of the L1 cache size? Hint: Start by expressing the global miss rate as a function of the L2 cache size.

GlobalMiss2 = 1-(L2/1MiB) = (1MiB-L2)/1MiB = L1/1MiB

4) What is the local miss rate for the L2 cache as function of the L1 and L2 sizes? Hint: Use your results from questions 2 and 3.

LocalMiss2 = [L1/1MiB] / [1 – (L1/1MiB)] = L1/(1MiB-L1) = L1/L2

5) Assume the hit time of the L1 cache is 10 cycles, the hit time of the L2 cache is 20 cycles and the memory access time is 100 cycles. Using the formula from question 1, what is the AMAT for this system as a function of only the L1 size?

AMAT = H1 + LocalMiss1 * (H2 + LocalMiss2 * Memory)
= 10 + [L2/1MiB]*[20 + (L1/L2)*100]
= 10 + (1MiB-L1)*20/1MiB + L1*100/1MiB
= 10 + 20 – 20*L1/1MiB + 100*L1/1MiB
= 30 + 80*L1/1MiB

6) What sizes of L1 and L2 caches should we pick to minimize the AMAT? (assume the caches have non-zero size, i.e., both of them exist)

L1 = 4B, L2 = 1MiB-4B
Question 3 (SU15 MT2):
Assume we are working in a 32-bit physical address space. We have two possible data caches: cache X is a direct-mapped cache, while cache Y is 2-way associative with LRU replacement policy. Both are 4 KiB caches with 512 B blocks and use write-back and write-allocate policies.

a) Calculate the number of bits used for Tag, Index and Offset:

<table>
<thead>
<tr>
<th>Cache</th>
<th>Tag bits</th>
<th>Index bits</th>
<th>Offset bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>20</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>Y</td>
<td>21</td>
<td>2</td>
<td>9</td>
</tr>
</tbody>
</table>

Use the code below to answer the following parts. Assume that ints are 4 B and doubles are 8 B.

```c
// 2^14 elements in the double array
int DOUBLE_ARRAY_SIZE = 2 * 1024;  // 2^14 bytes
double double_arr[DOUBLE_ARRAY_SIZE];

for (int i = 0; i < DOUBLE_ARRAY_SIZE; i++) /* loop 1 */
    double_arr[i] = i;
for (int i = 0; i < DOUBLE_ARRAY_SIZE; i += 8) /* loop 2 */
    double_arr[i] *= double_arr[0];
```

b) What is the hit rate for each cache if we run only loop 1? (hint: they’re both the same). What types of misses do we get?

**Both have a hit rate of 63/64. Compulsory**

c) What is the hit rate of each cache when you execute loop 2? Assume that you have executed loop 1. Assume the worst case ordering of accesses within a single iteration of the loop if multiple orders are possible. You may leave your answer as an expression involving products and sums of fractions.

X: ___See Below______________  Y: ____See Below__________

We are accessing with a stride of 8*8B = 64B while our block size is 512B. Thus, we have 8 accesses in each block. Since the array size is 2KiB*8B = 16KiB and our caches are 4KiB, the actual data is 4 times the size of our caches.

In cache X, the first quarter of the array will have a hit rate of 23/24 since we only encounter misses in each new block. In the rest of the array, however, we get a ping-pong effect on the first block as the loop requires double_arr[0]. For the 2nd-4th quarter of the array, the first block accesses will be (double_arr[i], double_arr[0], double_arr[i]):

Access 1: M, M, M
Access 2: H, M, M
This yields a hit rate of 7/24. The second through eighth block accesses have 23/24 hit rate.

Putting it all together, since we are accessing 4*8 blocks in total, hit rate = 29/32 * 23/24 + 3/32 * 7/24
Cache Y has a hit rate of 23/24 because there is no ping-pong effect in the first block.

d) Compute the AMAT for the following system with 3 levels of caches. (You should not need any information from the previous parts of this problem.) Give your answer as a decimal value.

<table>
<thead>
<tr>
<th>L1$</th>
<th>L2$</th>
<th>L3$</th>
<th>Main Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global miss rate: 50%</td>
<td>Local miss rate: 20%</td>
<td>Local miss rate: 1%</td>
<td>Hit time: 500ns</td>
</tr>
<tr>
<td>Hit time: 1ns</td>
<td>Hit time: 5ns</td>
<td>Hit time: 15ns</td>
<td></td>
</tr>
</tbody>
</table>

\[ 1 + 0.5 \times (5 + 0.20 \times (15 + 0.01 \times (500))) = 5.5 \text{ ns} \]