

Student Name: \_\_\_\_\_

Student ID: \_\_\_\_\_

1	2	3	Total

Marmara University - Faculty of Engineering  
Computer Engineering

Spring 2019-2020  
CSE 2138 - Systems Programming

Online Homework #3 - Makeup  
9 June 2020

I hereby swear that the work done on this online homework is totally my own; and on my honor, I have neither given nor received any unauthorized and/or inappropriate assistance for this homework. I understand that by the school code, violation of these principles will lead to a zero grade and is subject to harsh discipline issues.

Full Name: \_\_\_\_\_

Signature: \_\_\_\_\_

**Solution and Submission Notes:**

1. Book, slides and notes are open.
2. Internet usage is not allowed; however, computers and cell phones may be used only for viewing course materials provided for this course.
3. This online homework consists of 3 questions and 4 pages.
4. Please write your answers clearly and neatly to the appropriate place.
5. **Show all your work.**
6. Write your student ID, first name and last name, and total number of pages you used on top of each solution page.
7. If you have not printed this PDF file, write the sentences in the frame above and the date, in your handwriting, on top of the first page of your answer sheet, and sign it.
8. Write your solutions in your handwriting on blank A4 sheets.
9. Scan all the solution pages to a single PDF file, name it as "FirstName\_LastName\_ID.pdf". Upload your PDF file to <http://ues.marmara.edu.tr> before deadline.

1. (40 pts) Consider the following C code and the assembly code generated for it. Fill in the blanks with the appropriate expressions. (Note: you may only use symbolic variables a, b, c, and x from the source code in your expressions below - do not use register names.)

**Show all your work!**

<pre>movq    %rsp, %rbp movl    \$1, %eax cmpq    %rdx, %rdi jge     LBB0_2 notq    %rdi addq    %rdx, %rdi shrq    %rdi addq    \$2, %rdi movq    %rdi, %rax LBB0_2: popq    %rbp retq</pre>	<pre>long rec( long a, long b, long c){     if (_____)         return _____;     else{         long x = rec(_____, _____, c)+1;         return _____     } }</pre>
---	--

2. (30 pts) Assume that, your machine has the following properties:

- a 32KB L1 data cache and a 512KB L2 cache that are both cold initially,
- cache block size is 64B,
- if an L1 miss occurs, access to L2 cache takes 10 cycles
- if both L1 and L2 misses occur, access to memory takes 100 cycles
- integer size is 4B

Considering the following piece of code, calculate the total number of misses and total miss cycles for the given values of N.

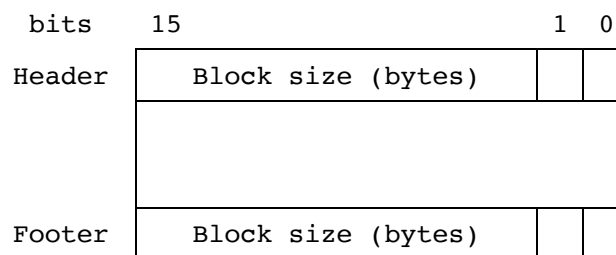
```
int a[N], k;
...
for (int i=0; i<N; i++)
    for (int j=0; j<N; j++)
        k = a[j];
```

- a) For  $N=2^{10}$ :    total number of L1 misses =  
                         total number of L2 misses =  
                         total miss cycles =
- b) For  $N=2^{15}$ :    total number of L1 misses =  
                         total number of L2 misses =  
                         total miss cycles =
- c) For  $N=2^{25}$ :    total number of L1 misses =  
                         total number of L2 misses =  
                         total miss cycles =

**Show all your work!**

3. (30 pts) Consider a memory allocator with the following properties:

- Allocations are 4-byte aligned, for each memory block, either allocated or free
- Allocator uses an implicit free list
- Allocator uses immediate coalescing, that is, adjacent free blocks are merged immediately each time a block is freed
- To find a free block, best fit method is used
- The allocated blocks have only header and free blocks have both header and footer
- The header and footer use:
  - the higher order 14 bits to record block size (block size includes the header and footer)
  - the lower order 2 bits to record allocation information:
    - bit 0 indicates the use of current block: 1 for allocated, 0 for free
    - bit 1 indicates the use of previous adjacent block: 1 for allocated, 0 for free.



If the heap starts at memory address 0xb000, show the contents of heap after each of the following requests. Note that the addresses grow from top down.

- Draw the heap
- Indicate the memory location that p1, p2, p3, p4, and p5 points at.
- Indicate each memory block either allocated or free.
- Indicate the blocks that contain header and footer.
- Give the contents of header and footer in hexadecimal for each block if exists. Don't forget to indicate the change in header and footer when there is a need.

- p1 = malloc(4)
- p2 = malloc(7)
- p3 = malloc(1)
- p4 = malloc(2)
- free(p1)
- free(p3)
- p5 = malloc(2)

**Show all your work and briefly explain your answer!**

Your heap will be like the following, and you will draw the heap 7 times (i.e., for each request). You can insert more rows as you need.

Address	Content	Header/Footer ?	is pointed ?
0xb000			
0xb002			
0xb004			
0xb006			
0xb008			
0xb00a			
0xb00c			

Here is an example that shows what you will include in your answer. **Note that this example is for a totally different allocator.**

Address	Content	Header/Footer ?	is pointed ?
0x400AFFC	0x00000013	Header	
0x400B000	Allocated		p1 points at this memory location
0x400b004	Allocated		
0x400b008	0x00000013	Footer	
0x400b00C	0x00000012	Header	
0x400b010	Free		
0x400b014	Free		
0x400b018	0x00000012	Footer	