

Worksheet 9

Due: April 18th (Wednesday) before 11:59 pm

Mode of submission: **canvas**

NOTE: There are **two (2)** questions in this worksheet worth a total of **20 points**.

1. Consider a dynamic memory allocator with the following properties. [8 points total]
 - **Double word** (8 bytes) aligned.
 - **Implicit free list** is used for free block organization.
 - All blocks have a **header** of size 4 bytes.
 - The **size** of a block (including the header) is stored in the header.
 - **bit 0** (least significant bit) in the header indicates the use of the current block:
 - 1 for allocated
 - 0 for free

Answer the following questions regarding this allocator:

- a. Minimum block size = _____
- b. Maximum block size = _____
- c. For the following memory allocation, what is the size of the payload and padding that will be used in the allocated block? Block splitting might happen based on the size of the request.

You should **assume** the following:

A free block of size **32 bytes** is chosen by the allocator to satisfy the below malloc request. The header of this block is at the memory location **0x8090A0B4**.

```
char *p = malloc(8);
```

Payload = _____ bytes

Padding = _____ bytes

Memory address stored in the pointer `p` (in hexadecimal):

0x_____

d. The contents of the header of a block in the allocator is **0x000000A9**.

- i. Is the block allocated or free? _____
- ii. What is the size of the block (in decimal)? _____
- iii. Is the contents of the header of this block valid with respect to this allocator? Remember, for a block to be valid with respect to an allocator, its size should satisfy the alignment requirement of the allocator.

YES (OR) NO

2. Consider an allocator with the following properties. [12 points total]

- **Single word** (4 bytes) aligned.
- **Implicit free list** is used for free block organization.
- Allocator uses a best-fit **policy** for allocating new blocks.
- All blocks have a header of size **4 bytes**.
- **bit 0** (least significant bit) in the header indicates the use of the current block:
 - 1 for allocated
 - 0 for free
- **block size** = sizeof (header) + sizeof (payload) + sizeof (padding)

Given the contents of the heap shown in Figure 1, show the new contents of the heap in FIGURE 2 **after** a call to **char *p = malloc(5);** is executed. Your answers should be given as **hexadecimal** values. If the value for a memory location in Figure 2 is the same as the value given in Figure 1, then you may just write **SAME** for the value in memory (in Figure 2).

What is the value of **pointer variable p** if the call to malloc succeeds? [1 point]

FIGURE 1
(BEFORE call to malloc)

Memory Address	Value in Memory
...	...
Start of heap => 0x8049000	0x00000009
0x8049004	0x12345678
0x8049008	0x00000010
0x804900c	0xA1B2C3D4
0x8049010	0x00000008
0x8049014	0x12CD00AB
0x8049018	0x00000008
0x804901c	0xFE670031
0x8049020	0x0000000c
0x8049024	0x00000004
0x8049028	0x00000002

FIGURE 2
(AFTER call to malloc)

Memory Address	Value in Memory
...	...
Start of heap => 0x8049000	
0x8049004	
0x8049008	
0x804900c	
0x8049010	
0x8049014	
0x8049018	
0x804901c	
0x8049020	
0x8049024	
0x8049028	