Q. Is there a Prelab we have to do?
- No prelabs in this course. But you are welcome to take a look at the lab handout (MP1_Handout.docx) so that we can have more interactive discussion during the lab.

Q. Anyone remember how to map network drive?
- Windows: https://wiki.cse.tamu.edu/index.php/Mapping_your_CSE_H:_Drive
- Mac OS: https://wiki.cse.tamu.edu/index.php/Mount_Your_Home_Directory_with_Mac_OSX

Q. So is my understanding that if the head pointer points to the first node, and the free pointer is moved to the end of that node by
```
free_pointer = head_pointer + (sizeof(head_pointer) * b);
```
If I print the addresses of both free_pointer and head_pointer, shouldn't there be a difference of b?
- The answer depends on the type of the variable head_pointer. In general the following is true:
```
type* ptr = addr;
ptr += x;  // advances ptr by x items, or x*sizeof (type) bytes
```

For instance, if you replace 'type' with 'char' in the above, you get:
```
char* ptr = addr;
ptr += x; // advances ptr by x bytes, because sizeof (char) = 1 byte
```

But when type is "struct node" you have:
```
struct node* ptr = addr;
ptr += x;  // advances ptr by x "struct node" items, or x*sizeof (struct node) bytes
```

So, if you know that you want to advance a pointer by some number of bytes (b in this case), then, you can typecast the type to (char*) and then advance it by b bytes and then typecast it back to the original type, like the following:
```
struct node* ptr = addr;
ptr = (struct node*)((char *)ptr +  x);  // advances ptr by x "struct node" items, or x*sizeof (struct node) bytes
```

Q. Do we check the value length against the actual value size or are we to assume that it will be the same?
- Remember we are not assuming anything about the data. So, it is not really possible to check the value size. You can safely assume that the caller of the Insert function is not lying about the length of the value. The only thing you should check is whether the value, given its length, will fit in the node.

Q. I am really confused about how to tell it to put the pointer first and then how to say the next four bytes are the key and how the next four are the length. How do you tell it where to store each variable?
- First, I assume that you have a correct structure defined. One example is:
```
typedef struct node {
        struct node* next;
        int key;
        int val_len;
} node_t;
```
Then, assume that you know the location to put this struct, which is say "addr". Now, you can do the following:

```
node_t * n = addr;
n->key = x;                     // places the key in memory address as the second field (leaving out space for the next pointer)
n->next = NULL;         // places NULL in addr, making the first few bytes (8 bytes) zero (i.e., point to nothing)
```

Q. Can I write functions for a struct?
- Yes, structs support member function

Q. What is the return Value of the LookUp function. The value or the key?
- Pointer to the key. But, as you already know that value len and the value immediately follow the key, having the key-pointer enables the caller to access those fields using pointer arithmatic as well.

Q. Do I keep a pointer to the value, or actually keep the value inside the node?
- You have to actually copy the input value onto the node. The function to use is 'memcpy()'

Q. Should both team-members submit the work?
- There is only submission required per team. So, only 1 member of a team should submit. The report should mention the names of the both members.

Q. I am trying to find a Segmentation Fault in my code. How to do that?
- You need to use a tool called 'gdb'. Here is how to use it:
http://cs.baylor.edu/~donahoo/tools/gdb/tutorial.html
Everybody, please get used to with gdb, this will be useful throughout the semester.

Q. In the Handout for MP1 it says something about write a program called testlist that takes input from command line what is this? and how do we do it?
- Testlist is the name of the executable. Its not a separate file or program. You just have to rename your program name from the default a.out to testlist when you compile. You also have to modify your main so that it accepts command line input. The following link will be helpful:
https://linuxprograms.wordpress.com/2012/06/22/c-getopt-example/