Some Practice Problems for Week 3

Think of the following problems **only after studying** the lecture-contents of this week.

Q1. During Week 2 you have seen how to implement a linked list in C. How can you make a linked-list more efficient by exploiting the memory hierarchy? For example, finding an element in the list becomes faster. There can be other operations as well.

Q2. What is a function pointer in C? Give an example where function pointer can be useful.

Q3. Are they the same or different?

```
int(*foo)(int);
int*foo(int);
```

Q4. The following C program verifies a password provided by a user. If the provided password matches with the stored password, then the program prints the contentsof a secret function. Otherwise the program terminates.

The program is running in a server and you have access to the program through a terminal. The program asks you to provide a 6-letter password. Your goal is to get inside the secret function.

You are aware of the source code, but you do not know what the secret password is. Describe a way to cheat the password verification scheme. [Hint: see buffer overflow]

```
#include<stdio.h>
#include<stdlib.h>
int secret_function(){
      printf("Inside secret function!\n");
      return 0;
}

int password_verify(){
      // Assume password is of length 6
      char received_password[7];
```

```c
	char password_stored[7]; // one extra for \0
	FILE *fp;

	// Program reads password from file
	fp = fopen("secret_file", "r");
	fscanf(fp, "%s", password_stored);
	fclose(fp);

	// Program receives user-input
	printf("Enter 6 letter password: ");
	scanf("%s", received_password);

	// Verify password char-by-char
	int i;
	for(i=0; i<6; i++){
		if(received_password[i] != password_stored[i]){
			printf("Password not matched\n");
			exit(-1);
		}
	}

	printf("Password matched! Welcome!\n");
	secret_function();
	return 0;
}

int main(){
	password_verify();
	return 0;
}
```