* 자료구조

1.
- 매일 진행되는 숙제를 좀 더 완벽하게 작성하도록 한다.
- 그날 배운 내용은 반드시 그 날 모두 이해하도록 한다.
- 그 주 주말에 한 주 동안 배운 내용을 다시 한번 복습하는 과정을 거친다.
- 학원에 좀 더 일찍 와서 공부하는 습관을 가진다.

2.
Q : 선입선출을 따르는 구조를 무엇이라 하는가?
A : 큐

3.
Q : 후입선출을 따르는 구조를 무엇이라 하는가?
A : 스택

4.
Q : 트리 자료구조는 연결 리스트에 비해 어떠한 이점을 가질수 있는가?
A: 스택, 큐와 같은 연결 리스트는 오로지 1차원으로만 자료가 나열된다. 반면에 트리 자료구조는 2차원적으로 자료가 나열되므로 좀 더 효율적으로 자료를 정리할 수 있다는 장점이 있다. 그 결과 자료의 검색, 삽입, 삭제가 연결리스트보다 빠르고 효율적이다.

5.
Q : AVL 트리와 일반 트리의 차이점이 무엇인가?
A : AVL 트리는 자료의 삽압과 삭제시 항상 양쪽 서브트리의 높이가 균형을 이루기 때문에, 어떤 자료든지 검색하는 시간이 가장 빨라진다.

6.
Q : 레드 블랙 트리는 왜 사용하는가?
A : AVL트리는 검색은 가장 빠른 트리이지만, 삽입과 삭제가 빈번하게 일어나는 경우 노드 하나하나마다 균형을 맞추어 주는 작업을 수행하기 때문에 굉장히 비효율적이다. 반면 레드 블랙 트리는 AVL 트리의 이러한 단점을 보완해서 삽입, 삭제 시 효율이 그렇게 떨어지지 않으면서도 검색이 양호한 실행속도를 가지므로 많은 웹 관련 회사에서 주로 사용한다.

* 임베디드 어플리케이션 분석

1.
- 매일 진행되는 숙제를 좀 더 완벽하게 작성하도록 한다.
- 그날 배운 내용은 반드시 그 날 모두 이해하도록 한다.
- 그 주 주말에 한 주동안 배운 내용을 다시 한번 복습하는 과정을 거친다.
- 학원에 좀 더 일찍 와서 공부하는 습관을 가진다.

2. int 형 변수 2개를 입력받아서, 합을 구하고 결과를 출력하는 프로그램.

```
#include <stdio.h>

void input(int *num1, int *num2);
int sum(int *num1, int *num2);
void print(int *sum);

int main(void)
{
	int num1, num2, sum1;
	input(&num1, &num2);
	sum1 = sum(&num1, &num2);
	print(&sum1);

	return 0;
}

void input(int *num1, int *num2)
{
	scanf("%d %d", num1,num2);	// int형 변수 2개를 입력받음
}

int sum(int *num1, int *num2)
{
	int sum = *num1 + *num2;	// 입력받은 2개의 변수의 합을 구함

	return sum;
}

void print(int *sum)
{
	printf("sum 은 %d 입니다.\n", *sum);  // 인자로 받은 sum 변수의 값을 출력함
}
```

3. c언어의 메모리 레이아웃은 4부분 나뉘어 진다.
- stack : 지역변수와 함수가 저장이 된다.
- heap : 동적 할당된 변수들이 저장된다.
- data : 전역변수와 static으로 선언된 변수가 저장이 된다.
- text : 기계어가 저장이 된다.

4. 함수포인터는 함수의 주소를 저장하는 변수이다. 함수포인터는 비동기 개념이다. 어떠한 상황이 발생했을 때, 어떤 행동을 하도록 되어있는 지침이라고 볼 수 있다. 함수포인터에 이러한 행동 지침을 미리 정의해 놓고, 그 상황이 닥쳤을 때 함수포인터를 호출해서 행동 지침을 수행하면 되는 것이다.

5. 함수를 여러개 만들지 않으면 main문 안에다가 모든 코드를 작성해야 한다. 그렇게 하면 main문 안이 복잡해져서 가독성이 떨어지고 SW의 전체 구성이 체계적으로 정리가 되지 않는다. 그렇기 때문에 각각의 기능별로 함수를 만들어서 정의해 놓고 main문 안에서는 해당 함수를 불러서 코드를 작성하면 되는 것이다.

6. 포인터는 변수의 주소를 저장할 수 있는 메모리 공간을 말한다. 포인터가 있음으로 인해서 전혀 다른 스택에 있는 변수를 컨트롤 할 수 있게 된다. 그렇게 함으로써 유연성이 생기게 된다.