

1. tms 사이트에 정답 입력
  2. pdf 혹은 notepad 제출
  3. 그림 그리는 것은 스샷 첨부
- [과목 문항 번호 표기]

모든 프로그래밍 문제는 반드시 기능별로 함수를 만들어서 작성하도록 한다.  
 함수 작성을 통해 기능을 분리하지 않을 경우 70% 감점  
 (통 함수, 통 메인 전부 감점 대상임)

[ 임베디드 애플리케이션 분석 ]

- 1/ 이것이 없으면 사실상 C 언어를 사용할 수 없다.  
 C 언어에서 함수를 호출하기 위해서도 이것은 반드시 필요하다.  
 이와 같은 이유로 운영체제의 부팅 코드에서도  
 이것을 설정하는 작업이 진행되는데 이것은 무엇일까 ?

main() 함수

- 2/ 배열에 아래와 같은 정보들이 들어있다.

```

2400, 2400, 2400, 2400, 2400, 2400, 2400,
1, 2, 3, 4, 5, 1, 2, 3, 4, 5,
2400, 2400, 2400, 2400, 2400, 2400, 1, 2, 3, 4,
5, 1, 2, 3, 4, 5, 2400, 2400, 2400, 2400, 2400, 5000,
1, 2, 3, 4, 5, 5000, 5000, 500, 500, 500, 500, 500,
1, 2, 3, 4, 5, 500, 500, 500, 500, 500, 500, 500, 500,
1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2,
3, 4, 5,
1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2,
3, 4, 5,
1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2,
3, 4, 5,
1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2,
3, 4, 5,
1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2,
3, 4, 5,
1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 7,
8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 234, 345, 26023, 346,
345, 234,
457, 3, 1224, 34, 646, 732, 5, 4467, 45, 623, 4, 356,
45, 6, 123,
3245, 6567, 234, 567, 6789, 123, 2334, 345, 4576, 678, 789, 1000,
2400, 2400, 2400, 2400, 2400, 2400, 1, 2, 3, 4, 5, 1,
2, 3, 4, 5,
2400, 2400, 2400, 2400, 978, 456, 234756, 5000, 5000, 5000, 2400,
500, 5000, 2400, 500,
500, 500, 500, 500, 500, 1, 2, 3, 4, 5, 500, 500,
500, 500, 500,
500, 500, 500, 500, 500, 1, 2, 3, 4, 5, 500, 500, 500,
5000, 2400, 5000,
5000, 5000, 5000, 5000, 5000, 5000, 5000, 1, 2, 3, 4, 5,
5000, 5000,
5000, 5000, 2400 5000, 500, 2400, 5000, 5000, 5000, 5000, 5000,
5000, 5000,
1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 5000, 5000,
5000, 5000, 5000,
1, 2, 3, 4, 5, 5000, 5000, 5000, 5000, 5000, 234,
4564, 3243, 876,
645, 534, 423, 312, 756, 235, 75678, 2400, 5000, 500, 2400, 5000,
500, 2400, 5000,

```

500, 2400, 5000, 500, 2400, 5000, 500, 2400, 5000, 500, 2400, 5000, 500,  
 2400, 5000, 500, 2400,  
 5000, 500, 2400, 5000, 500, 7, 8, 9, 6, 7, 8, 9, 6, 7, 8, 9, 6, 7, 8, 9, 6,  
 7, 8, 9, 6, 7, 8,  
 9, 6, 500, 2400, 5000, 500, 2400, 5000, 500, 2400, 5000, 500, 2400, 5000,  
 500, 2400, 5000,  
 500, 2400, 5000,

여기서 가장 빈도수가 높은 3 개의 숫자를 찾아 출력하시오!  
 함수에서 이 작업을 한 번에 찾을 수 있도록 하시오.  
 (찾는 작업을 여러번 분할하지 말란 뜻임)

3/ 12 비트 ADC 를 가지고 있는 장치가 있다.  
 보드는 12 V 로 동작하고 있고 ADC 는 -5 ~ 5 V 로 동작한다.  
 ADC 에서 읽은 값이 2077 일 때  
 이 신호를 디지털 관점에서 재해석하도록 프로그래밍 한다.

4/ 전세계 각지의 천재들이 모여서 개발하는 리눅스 운영체제 코드에는  
 엄청나게 많은 양의 goto 가 사용되고 있다.  
 goto 를 도대체 왜 사용해야만 할까 ?

goto문은 jmp 하나로 구성되었다  
 if 문은 기본적으로 mov, cmp, jmp로 구성  
 for 문의 경우 for 문이 여러개 생기면 if, break 조합의 경우  
 for 문의 갯수만큼 mov, cmp, jmp를 해야하는데 call이나 jmp를 분기명령어라고하는데  
 분기명령어는 파이프라인을 부순다.  
 가장 단순한 CPU가 실행까지 3 clock을 소요하는데  
 파이프라인이 깨지면 다시 3 clock을 버려야함을 의미  
 파이프라인 단계가 수십단계라면 분기가 여러번 발생하여 파이프라인 단계 x 분기횟수만큼  
 CPU clock을 낭비하게된다.

5/ 포인터 크기에 대해 알고 있는대로 기술하시오.

4 바이트

6/ TI Cortex-R5F Safety MCU is very good to Real-Time System.

위의 문장에서 Safety MCU 가 몇 번째 부터음  
 시작하는지 찾아내도록 프로그래밍 해보자.  
 (이정도는 가볍게 해야 파싱 같은것도 쉽게 할 수 있다)

7/ 이중 배열을 함수의 인자로 입력 받아 3 by 3 행렬의 곱셈을 구현하시오.

8/ void (\* signal(int signum, void (\* handler)(int)))(int) 의 프로토타입을 기술하시오.

리턴: void(\*) (int)

함수명: signal

인자: int signum 과 void (\*handler)(int)

9/ 함수 포인터를 반환하고 함수 포인터를 인자로 취하는 함수의 주소를 반환하고  
 인자로 int 2 개를 취하는 함수를 작성하도록 한다.  
 (프로토타입이 각개 다를 수 있으므로 프로토타입을 주석으로 기술하도록 한다)

10/ 파이프라인이 깨지는 경우 어떠한 이점이 있는지 기술하시오.  
 없음

11/  $4x^2 + 5x + 1$  을 1 ~ 3 까지 정적분하는 프로그램을 구현하도록 한다.

```
#include<stdio.h>
int main(void)
{
    int x;

    for(x=1;x<4;x++)

        printf("4x^2 + 5x + 1=%d\n", (4*x)^2+5*x+1);

    return 0;
}
```

12/ 함수 포인터를 활용하여 float 형 3 by 3 행렬의 덧셈과  
int 형 3 by 3 행렬의 덧셈을 구현하도록 하시오.

13/ 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... 형태로 숫자가 진행된다.  
1 ~ 27 번째까지의 수들로 홀수들의 합을 하고 짝수들의 합을 구한다.  
홀수들의 합 - 짝수들의 합의 결과를 출력하시오.  
(프로그래밍 하시오)

```
#include<stdio.h>

int main(void)
{
    int i;
    int sum1;
    int sum2;
    for(i=1;i<10000;i++){
        if(i%2==0)
            sum1+=sum1;
        else
            sum2+=sum2;
    }
    printf("짝수의 합 : %d\n",sum1);
    printf("홀수의 합 : %d\n",sum2);

    return 0;
}
```

14/ 1, 4, 5, 9, 14, 23, 37, 60, 97, ... 형태로 숫자가 진행된다.  
23 번째 숫자는 무엇일까 ?  
(프로그래밍 하시오)

15/ Intel Architecture 와 ARM Architecture 의 차이점은 ?

16/ 네이버의 사다리 게임을 우리끼리 즐길 수 있는 것으로 대체하는 프로그램을 만들어보자.  
우리반 학생들은 모두 25 명이다.  
반 학생들과 함께 진행할 수 있는 사다리 게임을 만들도록 한다.  
참여 인원수를 지정할 수 있어야하며  
사다리 게임에서 걸리는 사람의 숫자도 조정할 수 있도록 만든다.

17/ 아래와 같은 행렬을 생각해보자!

```
2   4   6
2   4   6
```

sapply(arr, func) 으로 위의 행렬을 아래와 같이 바꿔보자!

```
2   4   6
```

sapply 함수를 위와 같이 구현하라는 의미다.  
(R 이나 python 같은 언어에서 지원되는 기법중 하나에 해당한다)

18/ char \*str = "WTF, Where is my Pointer ? Where is it ?" 라는 문자열이 있다  
여기에 소문자가 총 몇 개 사용되었는지 세는 프로그램을 만들어보자

19/ int \*p[3] 와 int (\*p)[3] 는 같은 것일까 ? 다른 것일까 ?  
이유를 함께 기술하도록 한다.

다른 것, int \*p[3] : 크기 3인 포인터 배열 p  
int (\*p)[3] : 크기 3인 주소를 저장하는 포인터 변수 p

20/ 임의의 값 x가 있는데, 이를 134217728 단위로 정렬하고 싶다면 어떻게 해야할까 ?  
어떤 숫자를 넣던 134217728 의 배수로 정렬이 된다는 뜻임  
(힌트 :  $134217728 = 2^{27}$ )

21/ 단 한 번의 연산으로 대소문자 전환을 할 수 있는 연산에 대해 기술하시오.  
(프로그래밍 하시오), 덧셈 혹은 뺄셈 같은 기능이 아님

22/ 변수의 정의를 기술하시오.

변하는 수

23/ 포인터의 정의를 기술하시오.

주소를 저장할 수 있는 변수  
무언가를 가르키고 포인터의 크기는 HW가 몇 몇 bit를 지원하느냐에 따름

24/ 함수 포인터의 정의를 기술하시오.

함수를 저장할 수 있는 변수  
구조체 안에 함수를 집어넣을수 있는 걸 가능하게하는게 함수포인터

25/ 파이프라인은 언제 깨지는가 ?

함수가 호출되게하면 파이프 라인이 깨진다

26/ 메모리 계층 구조에 대해 기술하시오.

레지스터  
캐시  
메모리  
하드웨어

47/ C 언어의 기본 메모리 구조에 대해 기술하시오.

Stack : 지역변수가 위치하는 영역  
Heap : 동적 할당된 것들이 위치하는 영역  
Data : 전역변수 및 static으로 선언된 것들이 위치하는 영역  
초기화 되지 않은 모든것은 0으로 저장됨  
Text : Machine Code가 위치하는 영역

28/ 우리가 사용하는 윈도우나 리눅스, 맥(유닉스)에서 보는  
변수 주소는 진짜 주소일까 아닐까 ?  
알고 있는대로 기술하시오.

가짜주소, 이 주소값은 가상 메모리 주소에 해당하고 운영체제의 paging 매커니즘을 통해서  
실제 물리 메모리의 주소로 변환된다.

29/ 이름과 급여를 저장하도록 만든다.  
이름은 마음대로 고정시키도록 하고 급여는 rand() 를 활용  
급여의 평균을 출력하고 가장 높은 한 사람의 이름과 급여를 출력하시오.  
(값이 같을 수 있음에 유의해야 한다)

30/ 리눅스에서 디버깅을 수행하기 위한 프로그램 컴파일 방법을 기술하시오.

```
gcc -g ~.c
gdb a.out
b main
r //프로그램 실행
```

31/ 난수를 활용해서 Stack 을 구성한다.  
같은 숫자가 들어가지 않게 하고 20 개를 집어넣는다.  
이때 들어가는 숫자는 1 ~ 100 사이의 숫자로 넣는다.  
(마찬가지로 중복되지 않게 한다)

32/ 프로그램을 최적화하는 컴파일 옵션을 적고  
반대로 어떠한 최적화도 수행하지 않는 컴파일 옵션을 적도록 한다.

최적화 수행하는 컴파일 옵션 : shortcut  
최적화 수행않는 컴파일 옵션 :

33/ gdb 를 사용하는 이유를 기술하라.

디버깅은 컴파일은 성공적인데 논리적인 오류가 존재하는 경우에 수행하는 것  
그외에도 예측지 못한 문제로 무엇이 문제인지 파악하기위해 디버깅을 하는것

34/ 기계어 레벨에서 스택을 조정하는 명령어는 어떤것들이 있는가 ?

35/ a 좌표(3, 6), b 좌표(4, 4) 가 주어졌다.  
원점으로부터 이 두 좌표가 만들어내는 삼각형의 크기를 구하는 프로그램을 작성하라.

36/ 가위 바위 보 게임을 만들어보자.  
프로그램을 만들고 컴퓨터랑 배틀 한다.

37/ 화면 크기가 가로 800, 세로 600 이다.  
여기서 표현되는 값은 x 값이 [-12 ~ + 12] 에 해당하며 y 값은 [-8 ~ +8] 에 해당한다.  
x 와 y 가 출력하게 될 값들을 800, 600 화면에 가득차게 표현할 수 있는  
스케일링 값을 산출하는 프로그램을 작성하도록 한다.

38/ 등차 수열의 합을 구하는 프로그램을 for 문을 도는 방식으로 구현하고  
등차 수열의 합 공식을 활용하여 구현해본다.  
함수 포인터로 각각의 실행 결과를 출력하고  
이 둘의 결과가 같은지 여부를 파악하는 프로그램을 작성하라.

39/ sin(x) 값을 프로그램으로 구현해보도록 한다.  
어떤 radian 값을 넣든지 그에 적절한 결과를 산출할 수 있도록 프로그래밍 한다.

40/ 한 달간 수업을 진행하면서 본인이 느낀점을 20 줄 이상 최대한 상세하게 기술하시오.  
또한 앞으로에 대한 포부 또한 기술하길 바란다.  
그리고 앞으로 어떤 일을 하고 싶은지 상세히 기술하도록 한다.

첫날에는 우분투프로그램 설치하랴 진도따라가랴 정신이없었습니다  
결국 설치는 했지만 수업은 끝나있었고 그날부터 다시는 이런 불상사가없도록  
usb 를 지참하고 다니고있습니다  
그리고 뒷자리에 있다가 앞자리로 옮길수있는 기회가 생겨  
앞자리에서 수업을 잘들을수있게되었습니다  
셋째날에는 손필기가 느리다는걸 깨닫고 컴퓨터로 필기를 시작하였고

c를 배움에 있어 이상훈 강사님만큼 잘가르치는분은 일찍이 뵈적이 없어 진도를 잘못따라가는것에 대해 죄송한마음뿐이네요  
앞으로 더 열심히 하고싶고 잘따라가고싶습니다  
앞으로의 포부는 작다고 말할수도있지만 진도만 잘따라가고싶네요 :)  
방산업체에 취업을 하고싶은데 꼭 방산업체가 아니더라도 정부 산하기관도 생각하고있습니다  
둘 다 엄청난 경쟁률이라고 생각하고 열심히 하겠습니다 감사합니다^^

41/ 아래 Code를 작성하고 이 Code의 기계어에 대한 그림을 그리고 분석하시오.  
void swap(int \*a, int \*b){

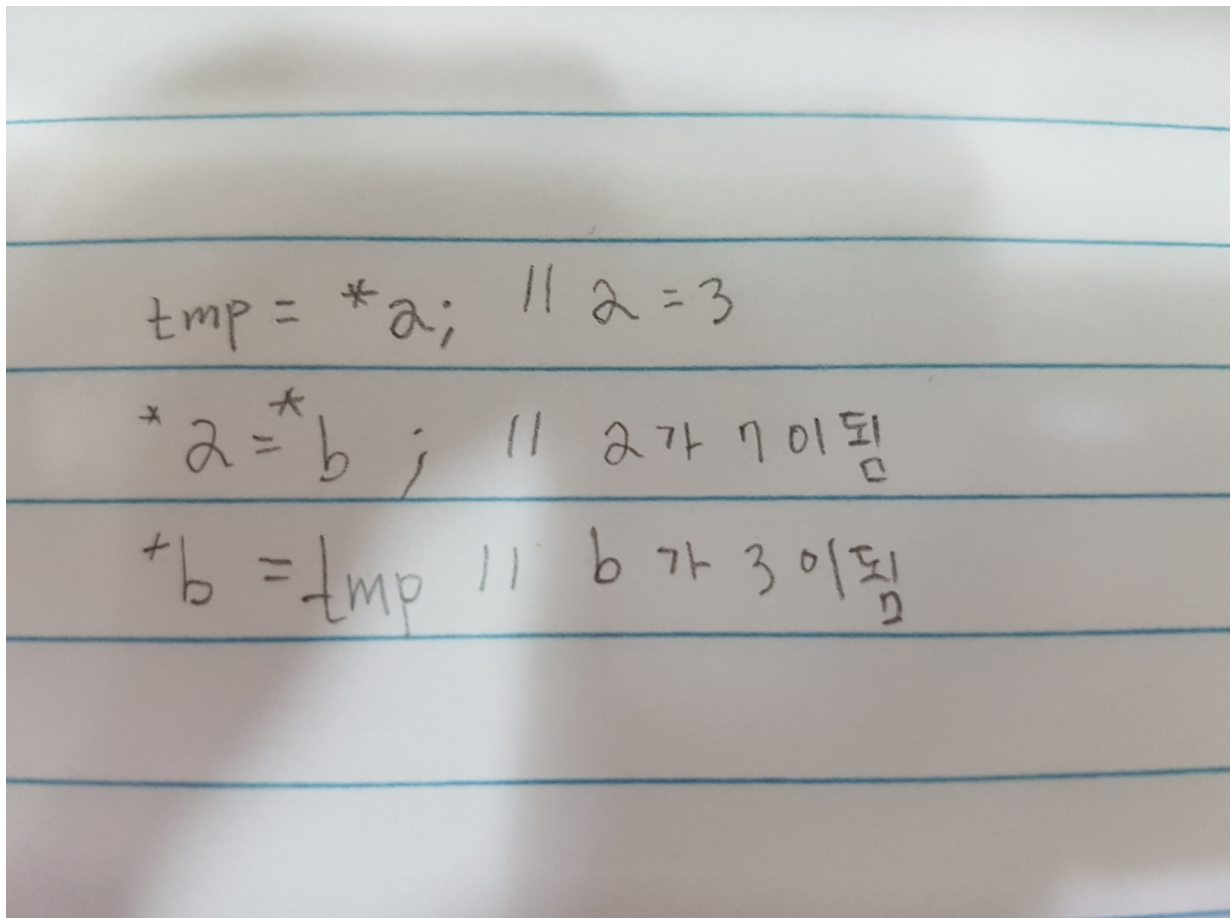
```
    int tmp;  
    tmp = *a;  
    *a = *b;  
    *b = tmp;
```

}

```
int main(void){
```

```
    int num1 = 3, num2 = 7;  
    swap(&num1, &num2);  
    return 0;
```

}



[ 자료구조 ]

1/ [복합문제 1.1] 값이 1 ~ 4096 까지 무작위로 할당되어 배열에 저장되도록 프로그래밍 하시오.  
내용 : 배열의 크기는 100 개 정도로 잡는다

2/ [복합문제 1.2] 앞서 만든 코드를 아래 설명을 보고 개량 하도록 한다.  
각 배열은 물건을 담을 수 있는 공간에 해당한다.

앞서서 100 개의 공간에 물건들을 담았는데 공간의 낭비가 있을 수 있다.

이 공간의 낭비가 얼마나 발생했는지 파악하는 프로그램을 작성하시오.

3/ [복합문제 1.3] 앞서 만든 코드를 보다 더 개량한다.  
문제에서 확장하여 공간을 보다 효율적으로 관리하고 싶어서 4096, 8192, 16384 등의 4096 배수로 크기를 확장할 수 있는 시스템을 도입했다.

이제부터 공간의 크기는 4096 의 배수이고 최소 크기는 4096, 최대 크기는 131072 에 해당한다.

발생할 수 있는 난수는 1 ~ 131072 로 설정하고 이를 효율적으로 관리하는 프로그램을 작성하시오.

(사실 리눅스 커널의 Buddy 메모리 관리 알고리즘임)

4/ 이진 트리를 재귀 호출을 사용하여 구현하도록 한다.  
재귀 호출로 구현한다.

```
#include<stdio.h>
```

```
typedef struct tree{
```

```
int data;
```

```
struct *link;
```

```
}tree;
```

```
int main(void)
```

```
{
```

```
return 0;
```

```
}
```

5/ 이진 트리를 재귀 호출 없이 구현하도록 한다.  
결과를 확인하는 print 함수(전위, 중위, 후위 순회) 또한 재귀 호출을 수행하면 안됨

6/ AVL 트리를 재귀 호출을 사용하여 구현하도록 한다.

7/ Red Black 트리와 AVL 트리를 비교해보도록 한다.

8/ 난수를 활용하여 Queue 를 구현한다.  
중복되는 숫자를 허용하지 않도록 프로그래밍 하시오.

제일 좋은 방법은 배열을 16 개 놓고 rand() % 16 을 해서 숫자가 겹치지 않는지 확인하면 된다.

9/ 재귀호출을 사용하여 queue 를 구현하고 10, 20 을 집어넣는다.

enqueue 과정에 대한 기계어 분석을 수행하여 동작을 파악하도록 한다.

그림과 함께 자세하게 설명하시오.

10/ [복합문제 2.1] 난수를 활용해서 Stack 을 구성한다.  
같은 숫자가 들어가지 않게 하고 20 개를 집어넣는다.

이때 들어가는 숫자는 1 ~ 100 사이의 숫자로 넣는다.

(마찬가지로 중복되지 않게 한다)

11/ [복합문제 2.2] 2.1 에서 만든 내용중 홀수만 빼내서 AVL 트리를 구성하도록 한다.

12/ [복합문제 2.3] 2.1 에서 짝수만 빼내서 RB 트리를 구성하도록 한다.

13/ 최적화 프로세스를 기술하도록 한다.

14/ 기존에는 숫자만 받아왔다.

이제 Queue 에서 데이터로서 숫자 값이 아닌 문자열을 받아보도록 하자.

15/ AVL 트리에 데이터로서 숫자가 아닌 문자열을 입력하도록 프로그램하시오.

16/ Binary Tree 에 문자열을 입력한다.

기존에 Data 에 숫자로 들어오던 정보에 문자열을 입력하도록 한다.

17/ 성적 관리 프로그램을 만들어보자.

여태까지 배운 학습 내용들을 활용하여 성적 관리 프로그램을 설계하고 구현해보자.

1. 통계 기능(총 합산, 평균, 표준 편차 계산)

2. 성적순 정렬 기능

3. 성적 입력 기능

4. 학생 정보 삭제 기능