

1.

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

void f(void);
double radio;
extern int counter;

int main(void){
    static int setting;
    register int index;
}

void f(void){
    int number;
```

2. 변수들의 범위, 생존기간, 연결을 설명하라

```
#include <stdio.h>

int a;
static int b;
extern int c;
int main(void){
    int d;
    register int e;
    static int f; {
        int g;
    }
    return 0;
}
```

변수 a는 전역 변수로 프로그램이 종료되기 전까지 메모리에 존재하며 모든 소스파일에서 사용가능하다. 변수 b는 static을 붙여 내부연결이 되며 하나의 소스파일에서만 사용가능하다. 변수 c는 extern을 사용하여 외부연결을 하며 모든파일에서 사용가능하다. d,e,f는 함수내부에서 사용되는 지역변수 이다. d는 레지스터변수로 레지스터에 할당되었다가 블록이 끝나면 (메인함수가 끝나면) 레지스터에 제거된다. 변수 f는 전역변수와 같이 프로그램이 실행을 종료하면 메모리에서 제거된다. 변수 g는 중괄호 안에서만 생존하게 된다.

3. 다음 프로그램의 출력을 쓰고, 이유를 설명하라.

(a)

```

#include <stdio.h>

void f(void);

int i;
int main(void){
    for (i = 0; i < 3; i++)
        f();
    return 0;
}
void f(void){
    for (i = 0; i < 5; i++)
        printf("#");
}

```

출력 : #####

메인함수에서 f()를 호출한 뒤 f()함수에서 i가 5가 되기 때문에 for문을 1번만 돌게 된다.(i 가 전역변수이기 때문에)

(b)

```

#include <stdio.h>

void f(int);

int n = 10;
int main(void){
    f(n);
    printf("n = %d\n", n);
    return 0;
}
void f(int n){
    n = 20;
}

```

출력 : n = 10

f(int n)에 있는 n은 f()함수에서만 쓰이는 지역변수이기 때문에 메인함수에 n값은 10으로 출력된다.

(c)

```
#include <stdio.h>

void f(void);
int x = 1;
int main(void){
    int x = 2;
    {
        int x = 3;
        printf("%d\\n", x);
    }
    printf("%d\\n", x);
    return 0;
}
```

출력: 3
: 2

중괄호안에서 출력하게 되는 x는 중괄호 안에만 생존하게 되는 x를 프린트 하기 때문에 3을 출력하고 메인함수 안에서 출력하는 함수는 메인함수 내에서 선언된 지역변수를 출력하기 때문에 2를 출력한다.

(d)

```
#include <stdio.h>

void f(void);
int main(void){
    f();
    f();
    return 0;
}

void f(void){
    static int count = 0;
    printf("%d\\n", count++);
}
```

출력 : 0
: 1

처음 f()를 호출하게 되면 처음 값인 0을 출력하게 되고 count는 출력 후 1이 된다. count는 static으로 선언되었기 때문에 f()함수로 다시 들어가도 0으로 초기화되지 않고 1을 출력한 뒤 count 값은 2가 된다.

4. 다음 소스에 오류가 있는지를 먼저 판단하고 오류가 있다면 오류를 지적하라. 논리적인 오류도 포함된다.

(a)

```
#include <stdio.h>

register auto int x = 20;

int main(void){
    printf("%d\n", x);
    return 0;
}
```

register 와 auto 변수는 지역변수이기 때문에 함수 내부에 선언하여야 한다.

(b)

```
#include <stdio.h>

int recursive(int n){
    if (n == 1) return 0;
    return n * recursive(n);
}
```

-> 함수는 순환구조로 사용하기 위해서는 return (n + recursive(n))을 return (n + recursive(n-1))로 고쳐 사용해야 순환구조로 작용한다.

위와 같이 하게되면 무한으로 n을 더하게 된다.

5. 다음 함수를 주석과 같이 호출하는 경우에 화면에 출력되는 내용과 함수의 반환값을 구하라

(a)

```
#include <stdio.h>

//sum(5)로 호출
int sum(int n){
    printf("%d\n", n);
    if (n < 1) return 1;
    else return (n + sum(n - 1));
}
```

출력 : 5 4 3 2 1 0

반환 : 1 6

(b)

```
#include <stdio.h>

//recursive(5)로 호출
int recursive(int n){
    printf("%d\\n", n);
    if (n < 1) return 2;
    else return(2 * recursive(n - 1) + 1);
}
```

출력 : 없음

반환 : 1 6

6.다음의 순환적인 프로그램을 반복구조를 사용한 비순환적 프로그램으로 바꿔라.

```
#include <stdio.h>

int recursive(int n){
    if (n == 0) return 1;
    else return(n + recursive(n - 1));
}
```

answer :

```
#include <stdio.h>

int recursive(int n){
    int sum = 0;
    while (n > 0){
        sum = sum + n;
        n -= 1;
    }
    if( n== 0) sum = sum + 1;
    return sum;
}
```