

1. 다음 소스의 각 출력 값을 적으시오.

```
int i=10, j=15, *p = &i, *q = &j, *r;  
char s1[] = "C programming is easy.";  
char *s2[] = {"C programming is easy.", "java", "c#"};
```

```
printf("%d\n", (*p)++);
```

```
printf("%d\n", i);  
printf("%d\n", (*(r = q) %= i));
```

```
printf("%d\n", j);  
printf("%d\n", sizeof(i)-sizeof(j));
```

```
printf("%d\n", strlen(s2[1]));
```

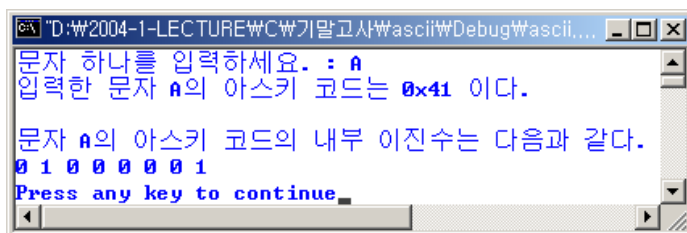
```
printf("%d\n", strcmp(s1, s2[1]));
```

```
printf("%s\n", (s1 + 2));
```

2. 다음은 무슨 프로그램인지 설명하고, 실행의 한 예를 들어 결과를 쓰시오.

```
#include <stdio.h>  
  
int main(void)  
{  
    int digit, n, sum = 0;  
    scanf("%d", &n);  
  
    do {  
        digit = n % 10;  
        sum += digit;  
        n /= 10;  
    } while(n > 0);  
  
    printf("%d\n", sum);  
}
```

3. 문자 하나를 입력 받아, 이 문자의 아스키 코드 값의 8개의 비트값을 모두 출력하는 프로그램을 아래와 같이 결과가 나오도록 프로그램 하시오.



4. 다음 프로그램의 결과를 기술하시오.

```
#include <stdio.h>
int main(void)
{
    short a[][3] = {2, 4, 6, 8, 10, 12, 1, 3, 5, 7, 9, 11};
    printf( "%d\n", sizeof(a) ); //배열 전체 크기
    printf( "%d\n", sizeof(a[0]) ); //1 행의 크기
    printf( "%d\n", sizeof(a[0][0]) ); //원소인 short 의 바이트 크기
    printf( "%d\n", sizeof(a) / sizeof(a[0]) ); //행의 수
    printf( "%d\n", sizeof(a[0]) / sizeof(a[0][0]) ); // 열의 수
    printf( "%d\n", sizeof(a) / sizeof(a[0][0]) ); //모든 원소의 수
    return 0;
}
```

5. 다음과 같이 일차원 배열을 복사하는 함수를 작성하여 이를 점검하는 프로그램을 작성하여 테스트 하시오.

```
copy_array(int from[], int to[], int n /* 배열 원소 수 */)
```

// 배열 from의 첫 번째 원소부터 n 번째 원소까지 같은 순서대로 배열 to로 값을 복사하는 함수

6. 2개의 소스로 구성된 다음 프로그램의 출력 결과를 쓰시오.

```
#include <stdio.h>

int a();
int g = 200;

void main(void)
{
    int b = 10;
    printf("%d\n", g + b) ;
    printf("%d\n", a() ) ;
    printf("%d\n", a() ) ;
    printf("%d\n", a() ) ;
}
```

```
static int sg = 100;

int a()
{
    static int s = 10 ;
    return s++ + sg++;
}
```

7. 다음 배열에서 모든 원소의 값을 모두 n씩 증가시키는 함수를 작성하고 테스트 하시오.

```
int data[] = {3, 21, 35, 57, 24, 82, 8};
```

8. 정수 100개를 저장할 공간을 동적 할당 받고, 랜덤 함수를 이용해 1 - 100 사이의 값 100개를 랜덤 생성하여 저장한 후, 생성된 100개 값의 최대, 최소, 평균, 분산을 출력하는 프로그램을 작성 하시오. 마지막에 반드시 동적 할당 받은 메모리를 반환할 것!