C프로그래밍1 멘토링 기말고사 대비

이름 _____

학번 _____

[Type A]

다음 질문에 답하시오.

1. 각 자료형에 맞는 바이트 크기를 쓰시오.

    int : [4] Byte = [32] Bit

    double : [8] Byte = [64] Bit

    char : [1] Byte = [8] Bit

    short : [2] Byte = [16] Bit

    float : [4] Byte = [32] Bit

2. C언어의 변수 선언 규칙에 위배된 것을 모두 고르시오.

    a) int count = 0;

    b) char _root = 'a';

    c) double 3PI = 3.141592;

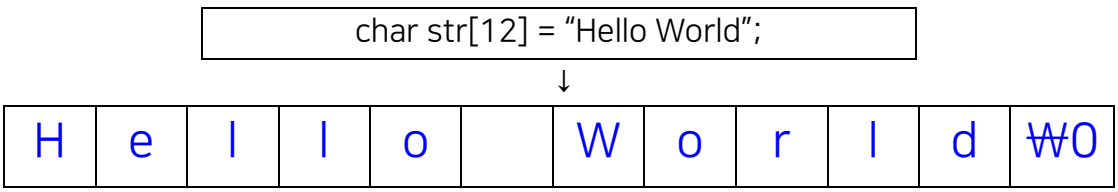
    d) int my number = 31;

3. 다음 2진수를 2의 보수법을 활용하여 10진수로 변환하여라.

    (1) 01001111 -> 79

    (2) 10101001 -> -87

4. 다음처럼 문자열을 선언하고 초기화했을 때, 메모리에 저장되는 결과를 쓰시오.

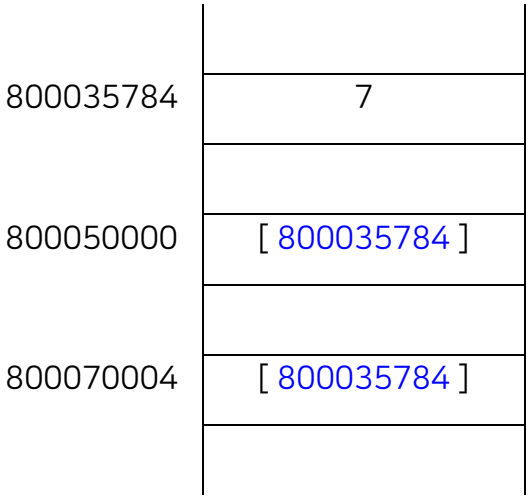


5. 왼쪽은 정수형 변수와 포인터형 변수를 선언 및 초기화한 코드이고, 출력되는 내용은 주석과 같다. 오른쪽은 대응되는 메모리 상태를 나타낼 때, 그림의 빈 칸을 채우시오.

(메모리 주소, 변수 값 및 출력은 모두 10진수라 가정한다.)

```
int main (void) {
    int num = 7;
    int* p1 = &num;
    int* p2 = p1;
    printf("%d, %d ₩n", &num, &p1);
    // 800035784, 800050000

    return 0;
}
```



6. 시스템의 주소 값 크기가 32비트인 시스템에서 int* 타입 변수의 크기는 [4] 바이트이며, double* 타입 변수의 크기는 [4] 바이트이다.

다음 문장에 대해서 맞으면 T, 틀리면 F를 쓰시오.

7. 배열의 이름은 배열의 시작 주소 값을 의미하는 포인터이다. 즉, 배열의 첫 번째 요소를 가리키는 포인터이다. (T / F)

8. break 문은 자신을 감싸는 반복문 한 개를 빠져나간다. (T / F)

9. 1 Byte는 8 Bit이며, 1 Byte로 표현할 수 있는 정보의 양은 2^8 개이다. (T / F)

10. 정수를 표현하는 데이터 타입의 크기가 1 Byte 라면, 음수와 양수를 표현할 수 있는 범위는 -2^7 - 1 ~ 2^7 이다. (T / F)

11. 실수형 자료형의 선택 기준은 정밀도이다. (T / F)

12. 대문자 A의 아스키 코드 값은 소문자 a의 아스키 코드 값보다 크다. (T / F)

13. double형 데이터 입력에 사용되는 서식문자는 %f이다. (T / F)

[Type B]

1. 아래 문자열의 출력 결과가 "I like C" 이기를 바란다. 빈 칸을 채우시오.

```
int main (void) {
    char str[50] = "I like C Programming";
    str[  8  ] = '₩0';   // 특정 요소에 널(NULL) 문자 저장
    printf("String : %s ₩n", str);
    return 0;
}
```

2. 별 찍기 코드이다. 다음 코드의 예상되는 실행 결과를 구하여라.

```
int main (void) {
    int num = 5;
    for(int line = 1 ; line <= num ; line++) {
        for(int blank = num – line ; blank > 0 ; blank--) {
            printf(" ");
        }
        for(int star = 1 ; star <= 2 * line – 1 ; star++) {
            printf("*");
        }
        printf("₩n");
    }
    return 0;
}
```

실행 결과 :

```
    *
   ***
  *****
 *******
*********
```

3. 사용자가 다음 코드를 실행하였는데, 문자 입력 기능이 제대로 수행되지 않았다. 문자를 입력 받지 못한 이유를 쓰고, 어떤 문장을 새로 추가하여 문자 입력 기능을 제대로 수행할 수 있을지 추가할 문장과 위치를 적어라. (ex) "n번째 줄과 n+1번째줄 사이에...")

```
1  int main (void) {
2      int num;
3      char ch;
4
5      printf("숫자를 입력해보세요. ₩n");
6      scanf("%d", &num);
7      printf("문자를 한 번 입력해보세요. ₩n");
8      scanf("%c", &ch);
9      printf("총 2번 입력 받을 수 있으셨나요? ₩n");
10
11     return 0;
12 }
```

이유 : 예시) 버퍼를 완전히 비워주지 않았기 때문이다.

추가할 문장과 위치 : 예시) 8번째 줄과 9번째 줄 사이에 getchar( ); 를 삽입한다.

4. 두 원의 반지름을 정수 값으로 입력 받은 후, 두 원 넓이의 합을 실수형으로 반환해주는 함수 TwoCircleSize를 완성하시오.

```
double TwoCircleSize ( int a , int b ) {
    return a*a*3.14 + b*b*3.14;
}
int main (void) {
    printf("%.2f ₩n", TwoCircleSize(1, 2));
}
```

5. 배열 arr의 모든 요소 값들 중 배열의 index가 0 또는 3의 배수인 요소 값의 합을 구하고자 한다. 배열을 가리키는 포인터 변수 p의 포인터 연산과 참조를 통해 프로그램을 완성하시오.

```
int main (void) {
    int arr[100];
    int sum = 0;
    int* p = arr (혹은 &arr[0])   // p가 배열의 시작 주소를 가리킴
    for (int i = 0; i < 100; i++)
        arr[ i ] = i; // 배열의 각 요소에 0~99 까지 대입

    for(int i = 0; i < 100; i += 3) { // 배열 요소 값의 합을 더함
        sum += *(p+i)   // p와 i를 이용하여 표기할 것.
    }
    printf("%d ₩n", sum);
    return 0;
}
```

6. 세 변수에 저장된 값을 서로 뒤바꾸는 함수를 다음과 같이 정의해보자. 함수 호출의 결과로 num1에 저장된 값은 num2에, num2에 저장된 값은 num3에, 그리고 num3에 저장된 값은 num1에 저장되어야 한다. 빈 칸을 채우시오.

```
void Swap3 (int* n1, int* n2, int* n3) {
    int temp = *n3;
    *n3 = *n2;
    *n2 = *n1;
    *n1 = temp;
}
int main (void) {
    int num1 = 1, num2 = 2, num3 = 3;
    Swap3(&num1, &num2, &num3);
    return 0;
}
```

7. 1부터 x까지 합을 구하는 함수는 $f(x) = \sum_{i=1}^{x} i = x + \sum_{i=1}^{x-1} = x + (x-1) + \sum_{i=1}^{x-2} = \cdots$ 이므로, 재귀적으로 정의될 수 있다. 이 함수를 표현한 재귀함수 코드를 완성하시오.

```
int RecursiveSum (int n) {
    if (n == 1) {
        return 1;
    }
    else {
        return n + RecursiveSum(n-1);
    }
}
int main (void) {
    printf("%d ₩n", RecursiveSum(5));
    return 0;
}
```

8. 아래 코드의 실행 결과에서 밑줄 친 빈 칸 부분을 채우시오.

```
int main (void) {
    int a[5] = {1, 2, 3, 4, 5};
    int* p = a + 1;
    p++;
    printf("%d %d %d ₩n", p, a, p[1]);
    return 0;
}
```

실행 결과 : 18349768 18349760 4