

## 2019년도 2학기 인문대생을위한창의적프로그래밍 기말고사 ( 1교시 )

학번: \_\_\_\_\_ 이름: \_\_\_\_\_

- 1교시는 여러분이 특정 목표를 달성하기 위한 적절한 C 코드 요소를 작성할 수 있는지를 주로 묻습니다. 여러분이 작성한 답안은 우리 수업 표준 환경을 기준으로 채점됩니다.
- 답안 칸 밖에 적은 글씨들은 절대 채점에 반영되지 않습니다(답안은 마련된 칸 안에 적어 주세요). 답안 칸 안에 적은 모든 글씨들은 채점에 반영됩니다(답안 칸에 이상한거 적지 말아 주세요). 답안 칸에 있는 글씨를 지울 수 없다면 그 글씨에 ~~가로줄을 두 개~~ 그어 주세요.
- 여러분이 적은 글자는 사람인 강사가 읽습니다. 모호한 글자를 적는 경우 다른 곳에 적은 동일한 글자들의 필체를 감안하여 인식을 수행할 예정이나, 여러분의 의도가 명백히 강사에게 전달되지 않을 가능성이 있습니다. 이 점을 감안하여 몇몇 혼동하기 쉬운 글자들은 꼭 또박또박 적어 주세요.
- 일부 문제는 답안 적을 곳을 두 개 마련해 두었습니다. 위 칸을 쓰다가 다시 적으려는 경우 줄 곳고 아래 칸을 사용해 주세요. 만약 두 칸 모두 글씨가 적혀 있다면 무조건 아래 칸의 내용을 답안으로 간주합니다.
- 1교시 시험은 60분 동안 진행되며 만점은 50점입니다.

(1 ~ 3) 다음 선언들을 먼저 읽고, 문항별 목표들을 달성할 수 있는 수식을 각각 적어 주세요.  
해당 목표를 달성하는 것이 불가능하다면 '불가능'이라 적어 주세요.

번호	내용
1	int number ;
2	int *ptr ;
3	int arr[3];
4	int func();

1. ptr에, number에 대해 정의된 위치 값(포인터 값)을 담는 수식 (2점, 2분)

2. ptr에, arr의 0번째 칸에 대한 위치 값(포인터 값)을 담는 수식 (3점, 2분)

3. number에, func의 return값의 크기(Byte 수)에 해당하는 값을 담는 수식 (3점, 2분)

( 다음 페이지에서 계속됩니다 )

(4 ~ 8) 다음 선언들을 먼저 읽고, 문항별 목표들을 달성하기에 가장 적합한 선언을 각각 적어 주세요. 해당 목표를 달성하는 것이 불가능하다면 '불가능'이라 적어 주세요.

번호	내용
1	int number ;
2	double *ptr ;
3	void func() ;

주의사항:

- 새 선언을 적을 때 새 이름은 자유롭게 정해도 좋아요.
- 표현 '가장 적합한'은 deduction을 고려하지 않음을 의미해요.
- 여기서는 initializer나 인수 형식은 신경쓰지 않아도 좋아요.

4. 위의 1번 선언에 대해 정의된 위치 값(포인터 값)을 담을 수 있는 변수 선언 (3점, 2분)

5. 위의 2번 선언에 대해 정의된 위치 값(포인터 값)을 각각 담을 수 있는 세 칸짜리 배열 선언 (4점, 3분)

6. 위의 3번 선언에 대해 정의된 위치 값(포인터 값)을 각각 담을 수 있는 세 칸짜리 배열 선언 (5점, 5분)

7. 수식 func()의 형식과 동일한 형식 값을 담을 수 있는 변수 선언 (4점, 4분)

8. 수식 ptr[1]의 형식과 동일한 형식 값을 return하는 함수 선언 (4점, 3분)

( 다음 페이지에서 계속됩니다 )

(9 ~ 10) 아래의 예시 코드 구조를 잘 보고, 문항별 주어진 목표를 달성하기 위한 함수 정의를 각각 적어 주세요.

```
// 여러분이 적는 함수 정의는 이 부분에 위치하게 될 예정

int main()
{
    int arr[2];

    arr[0] = put_five();
    fill_with_three(&arr[1]);

    return 0;
}
```

주의사항:

- 이번에는 약간 쉬운 문항이 나왔어요. 푼 다음 꼼꼼히 점검해 보고 다음 문제로 넘어가면 좋을 듯 해요.
- 함수 정의 밖에는 아무 것도 적으면 안 되고, 한 문항에 대해 함수 정의를 여러 개 적어도 안 돼요.
- 함수 정의 안에 **선언**도 문장도 아닌 다른 것(예: #include 등)을 적으면 0점으로 간주돼요.
- 제약 조건이 있음에도 성공적으로 목표를 달성할 수 있는지를 묻는 것이므로 위 내용을 꼭 지켜 주세요.

9. 목표: 5를 return하는 함수 put\_five() **정의**하기 (4점, 3분)

10. 목표: 인수로 받은 int **포인터** 값을 통해 그 칸(**포인터**를 '따라가면' 있는 int 칸)에 3을 담아 주는 함수 fill\_with\_three() **정의**하기(인수 이름은 자유롭게 정해요) (4점, 4분)

(11 ~ 12) 아래의 선언들 및 각 함수들에 대한 설명들을 잘 보고, 문항별 주어진 목표를 달성하기 위한 함수 정의를 각각 적어 주세요.

```
// 매 호출마다 사용자 입력의 '다음 숫자(0 이상, 256 미만)'를 return해 주는 함수입니다.  
// 단, '다음 숫자'가 더 이상 없다면 -1을 return하며, 사용자는 언제나 숫자를 하나 이상 입력할 거예요.  
int G();  
  
// 인수로 지정한 숫자를 출력해 주는 함수입니다.  
// 어떻게 출력하는지는 신경쓰지 않아도 좋아요.  
void P(int arg);
```

주의사항:

- 새로 정의하는 함수의 형식은 자유롭게 정해요.
- 여전히 함수 정의 안에 선언도 문장도 아닌 다른 것(예: #include 등)을 적으면 0점으로 간주되며, 위에 선언된 두 함수 외에 다른 함수에 대한 호출식을 적는 것 또한 불가능해요.

11. 목표: 사용자 입력의 모든 숫자들 중 가장 큰 숫자 하나를 출력하는 함수 func11\_max() 정의하기 (6점, 15분)

설명 부분만 복붙해 왔어요. 12번 문항을 풀 때 참고해 주세요.

```
// 매 호출마다 사용자 입력의 '다음 숫자(0 이상, 256 미만)'를 return해 주는 함수입니다.  
// 단, '다음 숫자'가 더 이상 없다면 -1을 return하며, 사용자는 언제나 숫자를 하나 이상 입력할 거예요.  
int G();  
  
// 인수로 지정한 숫자를 출력해 주는 함수입니다.  
// 어떻게 출력하는지는 신경쓰지 않아도 좋아요.  
void P(int arg);
```

12. 목표: 사용자 입력의 모든 숫자들 중 '단 한 번만 입력받은' 숫자들만 모두 출력하는 함수 func12\_unique() 정의하기 (8점, 15분)