

(C 프로그래밍) 기말고사

(시간: 18:10 ~ 19:30)

※ 주의 사항

- 소스파일 명은 자신의 **9자리학번.c**로 한다.
- 최종 파일은 **eCampus의 “과제함” - “기말고사”**에 제출한다.
- **전역 변수**를 사용하여서는 **안 된다**.
- 문제에서 주어진 코드는 **변경없이 그대로 기말고사 소스파일에 추가하여 사용**한다. 단, 문제 파일에서 코드를 **복사/붙여넣기 하지 않는다**. 인코딩 문제로 문법적 에러가 발생할 수 있으며, 이에 따른 오류로 실행되지 않는 경우에는 부분점수 없이 0점 처리한다.

※ 구현할 프로그램은 “1. a의 b승 구하기”과 “2. 배열 합병” 그리고 “3. 더하기야 놀자” 세 가지 기능으로 구성된다. 프로그램은 메뉴를 통해서 사용자 입력을 받고, 사용자가 선택한 기능을 수행하며, 사용자가 프로그램을 종료할 때까지, 반복적으로 수행된다. 다음의 각 문제에서 제시하는 요구 사항들을 모두 만족하도록 프로그램을 완성하시오. 각 문제에서 명시한 함수 이름과 요구사항은 반드시 문제에서 제시한대로 구현하되, 그 이외의 프로그램 실행에 필요한 추가 사항들은 자유롭게 구현한다.

1. 화면에 메뉴를 출력하는 **show_menu()** 함수를 구현하시오. 함수는 아래 보기와 같이 **메뉴만 출력**한다. show_menu() 함수에서 사용자 입력을 받지 않는다. (5)

```
1. a의 b승 구하기
2. 배열 합병
3. 더하기야 놀자
4. 끝내기
메뉴를 선택하시오:
```

2. 함수 **main()**은 사용자에게 메뉴를 보여주고 사용자 입력에 따라서 해당 기능을 수행하는 동작을 반복 수행한다. 메뉴가 출력되었을 때, **모든 종류의 정수를 다 입력할 수 있다고 가정**한다. 따라서, **사용자 입력은 정수로 처리**한다. 다음의 세부 요구사항을 만족하도록 main() 함수를 구현하시오.

(1) 사용자가 4를 입력하면 프로그램은 정상 종료한다. (5)

(2) 사용자가 1~4 이외의 값을 입력하면 다시 메뉴를 보여주고 재입력 받는다. (5)

```
1. a의 b승 구하기
2. 배열 합병
3. 더하기야 놀자
4. 끝내기
메뉴를 선택하시오:6
```

```
1. a의 b승 구하기
2. 배열 합병
3. 더하기야 놀자
4. 끝내기
메뉴를 선택하시오:
```

- (3) 사용자가 1을 입력하면, **cal_pow()**함수가 호출되고, 2를 입력하면 **merge_array()**가 호출되고, 3을 입력하면 **play_add()**함수가 호출될 수 있도록 **switch 구문**을 이용하여 작성하시오. (5)
- (4) 사용자가 4을 입력할 때까지, 프로그램은 반복 수행된다. (5)
- (5) 그 외, 프로그램이 정상 수행되도록 코드를 완성한다. (5)

3. 사용자가 메뉴에서 1를 선택하면, 함수 **cal_pow()**가 호출된다. **cal_pow()** 함수는 임의의 두 개의 실수 a와 b를 입력받고, a의 b승을 계산하여 출력한다. **cal_pow()** 함수가 다음과 같이 주어졌을 때, **재귀 알고리즘**을 이용하여 재귀를 수행하는 **rpow(double, double)** 함수를 작성하시오. 반드시, 다음의 요구사항을 만족하도록 각 함수를 작성한다. 단, 지수는 정수만 입력한다고 가정한다.

[함수 정의]	
<pre>void cal_pow() { double a, b; printf("양의 실수 두 개를 입력하시오: "); scanf("%lf %lf", &a, &b); printf("%.2f의 %.2f승은 %.2f입니다. \n", a, b, rpow(a, b)); }</pre>	
[반복문을 이용한 재귀 구하기]	[실행 예]
<pre>int main() { double a, b; printf("실수 입력 (밑,지수): "); scanf("%lf%lf", &a, &b); double result = 1; for (int i = 0; i < b; i++) { result *= a; } printf("결과는 %lf", result); }</pre>	1. a의 b승 구하기 2. 배열 합병 3. 더하기야 놀자 4. 끝내기 메뉴를 선택하세요: 1 양의 실수 두 개를 입력하시오: 2 4 2.00의 4.00승은 16.00입니다.

- (1) 재귀 알고리즘을 기반으로 a의 b승을 구하는 재귀함수인 **rpow(double, double)**를 구현하시오. 반드시 재귀 함수로 구현한다. (20)

4. 사용자가 메뉴에서 2를 선택하면, 함수 `merge_array()`가 호출된다. 오름차순으로 정렬된 두 정수형 일차원 배열 `a1,a2`을 합병하고 다시 오름차순으로 정렬된 배열 `m`을 구한다. `merge_array()` 함수가 다음과 같이 주어졌을 때(기말고사 소스코드에 주어진 코드 그대로 추가), 다음의 요구사항에 맞게 사용자 정의 함수를 추가 구현하시오. 각 함수의 인자는 주어진 코드에 맞게 적절하게 정의하여 사용한다.

[함수 정의]
<pre> void merge_array() { int a1[ASIZE] = { 13, 27, 29, 54, 100, 231, 244, 350, 351, 459 }; int a2[ASIZE] = { 2, 3, 10, 13, 27, 141, 142, 190, 245, 400 }; // m 선언 printf("\n배열 합병 문제입니다. \n"); printf("첫 번째 배열 리스트: "); print_array_list(a1, ASIZE); printf("\n"); printf("두 번째 배열 리스트: "); print_array_list(a2, ASIZE); printf("\n"); merge(m, a1, a2); printf("합병된 배열 리스트: "); print_array_list(m, ASIZE * 2); } </pre>
[실행 예]
<pre> 1. a의 b승 구하기 2. 배열 합병 3. 더하기야 놀자 4. 끝내기 메뉴를 선택하세요: 2 배열 합병 문제입니다. 첫 번째 배열 리스트: 13 27 29 54 100 231 244 350 351 459 두 번째 배열 리스트: 2 3 10 13 27 141 142 190 245 400 합병된 배열 리스트: 2 3 10 13 13 27 27 29 54 100 141 142 190 231 244 245 350 351 400 459 1. a의 b승 구하기 2. 배열 합병 3. 더하기야 놀자 4. 끝내기 메뉴를 선택하세요: </pre>

- (1) 일차원 배열 `a1,a2`의 크기에 대한 기호 상수 `ASIZE`를 정의한다. (4)
- (2) 두 배열을 합병한 결과를 담은 변수 `m`을 동적할당하고, 0으로 초기화한다.(5)
- (3) `print_array_list()` 함수는 배열의 원소를 출력하는 함수이다. 문제에서 주어진 [실행 예]와 같이 출력되도록 함수를 작성하시오. (5)
- (4) `merge()` 함수는 두 배열을 합병하고 모든 원소들을 오름차순으로 정렬시켜 한 배열에 담은 함수이다. 문제에서 주어진 [실행 예]와 같이 출력되도록 함수를 작성하시오. (11)

- (5) **merge()** 함수의 인자는 모두 포인터로 받는다. (2)
- (6) **merge()** 함수에서 배열을 합병하는 과정에서 포인터를 사용한다. (10)

5. 사용자가 메뉴에서 3를 선택하면, 함수 **play_add()** 함수가 호출된다. 함수가 실행되면 사용자로부터 사용할 숫자 범위의 최소값과 최대값을 입력받고, 5번의 반복문이 실행된다. 한번의 반복마다 입력한 숫자 범위에서 **set_rand()** 함수를 통해 두 정수를 생성하고, "a + b = "의 형태로 문제를 출력하고 정답을 입력받는다. 정답을 맞추면 총 점수에 10점이 추가된다. 5개의 문제가 끝나면 총 수행시간과, 문제당 평균 시간, 이번 게임에서의 점수, 프로그램 실행동안의 총 점수가 출력된다. 다음의 요구사항에 맞게 구현하시오.

- (1) 숫자 범위의 최소값과 최대값을 입력 받는다. (2)
- (2) main() 함수에서 프로그램안에서의 총 점수인 total_score 변수를 선언하고 play_add()에서 그 변수를 참조한다. (4)
- (3) 인자로 입력받은 두 정수 사이의 범위 안에서 무작위의 정수를 반환하는 **set_rand(int, int)** 함수를 구현한다. (4)
- (4) 반복문을 통해 문제를 출력하고 정답을 입력 받는다. (6)
- (5) 게임 시작부터 끝까지 시간을 측정한다. (6)
- (6) 게임이 끝나고 총 수행시간, 문제당 평균 시간, 게임에서의 점수, 프로그램에서의 총 점수를 출력한다. (6)

[함수 정의]
<pre>void play_add(int* total_score) { } </pre>
[실행 예]
<pre> 1. a의 b승 구하기 2. 배열 합병 3. 더하기야 놀자 4. 끝내기메뉴를 선택하세요: 3 더하기야 놀자 게임을 시작합니다. 게임에 사용할 숫자 범위의 최소값과 최대값을 입력하세요: 2 10 7 + 10 = 17 9 + 6 = 15 10 + 3 = 13 5 + 2 = 7 9 + 4 = 13 총 수행 시간: 8초, 문제당 평균 시간: 1.60초 이번 게임에서 50 점을 획득하여서, 현재까지 총 점수는 50점 입니다. 1. a의 b승 구하기 2. 배열 합병 3. 더하기야 놀자 4. 끝내기 메뉴를 선택하세요: </pre>

★ 수고 많이 했습니다.