방학특강 후속과제 리스트

번호	번호	등급	
1. 순환문	1.2 삼각형 출력	기초1	
2. 함수	2.1 함수 생성 및 활용		
3. 배열	3.1 1차원 배열 : 생성 및 원소 값 표시	기초2	
	3.2 1차원 배열 : 평균/총점 구하기		
	3.3 1차원 배렬 : 원소 역방향 출력		
	3.4 1차원 배열 : 최소값과 최대값 찾기		
	3.5 2차원 배열 : 생성 및 원소 값 표시		
	3.6 행렬의 합	기초3	
	3.7 전치행렬		
4. 포인터	4.1 포인터 : 기본		
	4.1.1 변수의 주소와 포인터와의 관계	응용1	
	4.1.2 배열과 포인터		
	4.1.3 포인터를 이용한 배열 원소 인쇄		
	4.1..4 포인터를 함수 입력 인자로 전달		
	4.2 포인터 : 성적처리	응용2	
	4.3 포인터 : 행렬 곱		
	4.4 포인터 : Linear queue		
	4.4.1 구조체 멤버를 표현하는데 있어 포인터를 사용하는 버전		
	4.4.2 구조체 멤버를 표현하는데 있어 구조체 변수를 사용하는버전		
5. 파일	5.1 파일 생성 및 활용	응용3	

▶ 모든 문제는 실제 구현 각 문제에 대해 모두 보고서를 제출 (하나의 보고서에 모두 포함)

- 보고서에는 <소스코드>와 <결과 화면>이 포함되어 있어야 함
- 프로그램을 작성/구현하는 것도 중요하지만 이해를 완벽히 할 수 있도록 해야 함
- 제출 시 소스코드에 대한 이해도를 각 지도교수님이 확인할 예정임
- 필요시 소스코드에 주석을 달아도 무방함

■ 제출일: 개강후 첫 주 (9월 8일)내에 각 지도교수님께 제출하여 완료여부를 확인 받음

C 언어 교육자료 : 문제			
분야	C Programming	제목	1. 순환문
수준	기본		
작성일			

1.1 정수 처리

■ 문제설명

- 1 ~ 1000사이 정수를 대상으로 다음과 같은 실행결과가 나오도록 프로그램을 작성하시오

■ 실행결과

```
1 ~ 1000사이 숫자에 대해서 4로 나눈 나머지가 0인 숫자의 합은 125500
1 ~ 1000사이 숫자에 대해서 4로 나눈 나머지가 1인 숫자의 합은 124750
1 ~ 1000사이 숫자에 대해서 4로 나눈 나머지가 2인 숫자의 합은 125000
1 ~ 1000사이 숫자에 대해서 4로 나눈 나머지가 3인 숫자의 합은 125250
위 값들의 총합은 500500
```

1.2 삼각형 출력

■ 문제 설명

- 다음 실행결과와 같은 모양으로 출력하는 프로그램을 작성하시오.
  1. 대문자 A ~ Z의 ASCII 값은 65 ~ 90임.
  2. 전체 넓이는 글자 27자임.

ASCII(아스키, American Standard Code for information interchange) : 1963년 미국표준협회에 의해 결정되어 미국의 표준 부호가 되었다. 아스키는 128개의 가능한 문자조합을 제공하는 7비트(bit) 부호로, 처음 32개의 부호는 인쇄와 전송 제어용으로 사용된다. 보통 기억장치는 8비트(1바이트, 256조합)이고, 아스키는 단지 128개의 문자만 사용하기 때문에 나머지 비트는 패러티 비트나 특정문자로 사용된다. 1 ~ 9는 48 ~ 57이며, A ~ Z는 65 ~ 90이다.

■ 실행결과

```
                                                          A
                                                         AB
                                                        ABC
                                                       ABCD
                                                      ABCDE
                                                     ABCDEF
                                                    ABCDEFG
                                                   ABCDEFGH
                                                  ABCDEFGHI
                                                 ABCDEFGHIJ
                                                ABCDEFGHIJK
                                               ABCDEFGHIJKL
                                              ABCDEFGHIJKLM
                                             ABCDEFGHIJKLMN
                                            ABCDEFGHIJKLMNO
                                           ABCDEFGHIJKLMNOP
                                          ABCDEFGHIJKLMNOPQ
                                         ABCDEFGHIJKLMNOPQR
                                        ABCDEFGHIJKLMNOPQRS
                                       ABCDEFGHIJKLMNOPQRST
                                      ABCDEFGHIJKLMNOPQRSTU
                                     ABCDEFGHIJKLMNOPQRSTUV
                                    ABCDEFGHIJKLMNOPQRSTUVW
                                   ABCDEFGHIJKLMNOPQRSTUVWX
                                  ABCDEFGHIJKLMNOPQRSTUVWXY
                                 ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

C 언어 교육자료 : 문제			
분야	C Programming	제목	2. 함수
수준	기본		
작성일			

2.1 함수 생성 및 활용

■ 문제 설명

화면에서 숫자를 입력을 받아 아래와 같은 형식의 출력을 산출하는 프로그램을 함수를 사용하여 작성하시오.

▶ 프로그램의 함수 구성

- 메인 함수
- 두수를 입력을 받아 그 안에 있는 홀수(시작값 포함)의 합계를 내는 함수 : void odd_sum(int x, int y)
- 두수간의 홀수를 "(5, 7, 9)"와 같은 형식으로 출력하는 함수 : void odd_print(int x, int y)

▶ "다시 하겠읍니까"에서 "Y" 또는 "y"를 누르면 처음부터 다시 반복 수행함.

■ 실행결과

```
==== 1회 =====
1부터 20사이의 숫자를 입력하세요 : 5 10

<계산결과>
5 ~ 10까지의 숫자 중에서
홀수는 3개 (5, 7, 9)로서 합은 21 입니다.

다시 하시겠읍니까? (Y/N)  Y

==== 2회 =====
1부터 20사이의 숫자를 입력하세요 : 5 12

<계산결과>
5 ~ 10까지의 숫자 중에서
홀수는 4개 (5, 7, 9, 11)로서 합은 32 입니다.

다시 하시겠읍니까? (Y/N)  n

==== 2회 반복후 프로그램이 종료하였읍니다. =====
```

C 언어 교육자료 : 문제			
분야	C Programming	제목	3. 배열
수준	기본		
작성일			

3.1. 1차원 배열

■ 문제설명

- 크기가 5인 1차원 정수배열을 정의하고 아래와 같은 내용을 프로그램을 작성
  - 각 배열의 값을 원소의 인덱스 (0 ~ 4)의 10배로 초기화 선언
  - 초기 배열을 값을 인쇄
  - 프로그램에서 각 원소의 값에 인덱스의 값을 더한 후 배열의 값을 인쇄
  - FOR 문 사용

■ 실행화면

<초기값>

X[0]=0, X[1]=10, X[2]=20, X[3]=30, X[4]=40

<수정값>

X[0]=0, X[1]=11, X[2]=22, X[3]=33, X[4]=44

3.2 1차원 배열 : 평균/총점

■ 문제설명

- 5명의 학생들 성적을 입력받아 이의 총점과 평균 구하는 프로그램
- 크기가 5인 1차원 정수배열을 정의하며, 입력을 5개 받아 저장한 후 총점과 평균을 구함

■ 실행화면

실행결과

```
입력(점수 1) =>100↵
입력(점수 2) =>90↵
입력(점수 3) =>80↵
입력(점수 4) =>70↵
입력(점수 5) =>60↵
total= 400, avg= 80
```

3.3 1차원 배열 원소 역방향 출력

■ 문제설명

- 입력된 문자열을 역방향으로 출력하는 프로그램
- 1차원 배열을 정의한 후, 문자열을 받아 저장함
- 역방향으로 인쇄함

■ 실행화면

실행결과

```
Input String: seoul
Inverse String: luoes
```

3.4 1차원 배열에 저장된 숫자 중에서 최대값과 최소값을 찾는 문제

■ **문제설명**

◆ 아래 결과화면에서 보는 바와 같은 1차원 배열 선언후 최대값과 최소값을 찾는 문제

■ **실행 화면**

실행결과

```
배열의 값[ ]={ 76, -59, -34, 90, 169, -12, 13, 103, 3, 29}
최대값 ==> 169, 위치 ==> 5번째
최소값 ==> -59, 위치 ==> 2번째
```

3.5 2차원 배열

■ **문제설명**

◆ 3 x 4 크기를 갖는 2차원 정수(예 : X[3][4])에 대하여 아래 프로그램을 작성

- 각 배열을 값을 인덱스값을 나타내도록 (0, 1, 2, 3, 10, 11, 12, 13, ...) 초기화 선언, 값 인쇄
- 행의 순서 (0행 -> 2행, 1행 ->1행, 2행, 0행)가 바뀌도록 X배열의 값을 Y배열에 복사 값을 변경 및 인쇄

◆ FOR 문 사용

■ **실행화면**

<초기값>

X[0][0]= 0, X[0][1]= 1, X[0][2]= 2, X[0][3]= 3

X[1][0]=10, X[1][1]=11, X[1][2]=12, X[1][3]=13ㄱ

X[2][0]=20, X[2][1]=21, X[2][2]=22, X[2][3]=23

<수정값>

Y[0][0]=20, Y[0][1]=21, Y[0][2]=22, Y[0][3]=23

Y[1][0]=10, Y[1][1]=11, Y[1][2]=12, Y[1][3]=13

Y[2][0]= 0, Y[2][1]= 1, Y[2][2]= 2, Y[2][3]= 3

3.6 행렬의 합

■ **문제 설명**

▶ 3×3 행렬 2개를 읽어 들여 행렬 A, B에 저장하고 그 합을 행렬 C에 계산하여 저장한 후 각각 출력하는 프로그램

■ **실행 결과**

실행결과

```
*** Input the first matrix (3*3) ***
9↵
8↵
7↵
6↵
5↵
4↵
3↵
2↵
1↵
          ## A ##
9 8 7
6 5 4
3 2 1
*** Input the second matrix (3*3) ***
1↵
2↵
3↵
4↵
5↵
6↵
7↵
8↵
9↵
          ## B ##
1 2 3
4 5 6
7 8 9
*** C=A+B ***
          ## C ##
10 10 10
10 10 10
10 10 10
```

3.7 전치행렬

■ **행과 열을 바꾸는 프로그램 작성 (전치행렬) (단 X[5][5], Y[5][5])**

◆ 5 x 5 크기를 갖는 2차원 정수(예 : X[5][5])에 대하여 아래 프로그램을 작성

■ **실행 결과**

실행결과

```
원래의 행렬
4 5 7 8 9
5 2 4 6 2
4 7 8 5 2
3 5 8 9 6
7 5 6 9 8

전치 행렬
4 5 4 3 7
5 2 7 5 5
7 4 8 8 6
8 6 5 9 9
9 2 2 6 8
```

C 언어 교육자료 : 문제			
분야	C Programming	제목	4.1 포인터 : 기본
수준	기본		
작성일			

4.1.1 변수의 주소와 포인터와의 관계

■ **문제설명**

- 변수의 주소와 포인터와의 관계를 이해하기 위한 프로그램임
- 정수형 변수의 X, Y를 각각 10, 20으로 초기화한 후 포인터 P, Q가 각각 X와 Y를 가리키도록 한 후 다음과 같이 인쇄되도록 함

■ **실행화면**

```
변수 X의 값     X    = <값>
변수 Y의 값     Y    = <값>
변수 X의 주소   &X   = <값>
변수 Y의 주소   &Y   = <값>

포인터 P의 값   P  = <값>
포인터 Q의 값   Q  = <값>
포인터 P의 주소 &P   = <값>
포인터 Q의 주소 &Q   = <값>
P의 포인팅 값   *P   = <값>
P의 포인팅 값   *Q   = <값>
```

4.1.2 배열과 포인터

■ **문제설명**

- 10개의 크기를 갖는 1차원 정수 배열 A (초기값 0, 1 ... 9)를 포인터를 사용하여 원소의 주소와 원소값을 인쇄하는 프로그램 작성. (포인터 P가 A를 가리키게 한후 FOR 루프를 사용하여 구현)

■ **실행화면**

```
*P       : &A[0] = <주소>, A[0] = 0
*(P+1)   : &A[1] = <주소>, A[1] = 1
             ...
```

4.1.3 포인터를 이용한 배열 원소 인쇄

■ **문제설명**

- 4 x 3 크기를 갖는 2차원 정수(예 : X[4][3])에 대하여 원소값을 인쇄하는 프로그램을 작성
- 각 배열의 원소 값은 인덱스 값을 나타내도록 {{0, 1, 2}, {10, 11, 12}, {20, 21, 22}, {30, 31, 32}} 초기화 선언
- 포인터 P를 사용하여 FOR 루프를 2개 (외부, 내부) 사용하여 작성

■ **실행화면**

```
*(P+0) X[0][0]=  0,   *(P+1)  X[0][1]=  1,   *(P+2) : X[0][2]=  2,
*(P+3) X[1][0]= 10,   *(P+4)  X[1][1]= 11,   *(P+5) : X[1][2]= 12,
```

4.1.4 포인터를 함수 입력 인자로 전달

■ 문제설명

- 2개의 1차원 배열 A[]와 B[]의 내용을 서로 교환하는 프로그램

■ 프로그램 구조

```
1:  // 프로그램 헤더 : #include 문 등
2:
3:  swap(char *x, *y)
4:  {
5:    // 내용 추가
6:  }
7:
8:  void main()
9:  static char a[]="SEOUL";
10:   static char b[]="KOREA";
11:   printf("<교환전>₩n");
12:   printf("a[]= %s ₩n");
13:   printf("b[]= %s ₩n ₩n");
14:   swap(a, b);
15:   printf("<교환후>₩n");
16:   printf("a[]= %s ₩n");
17:   printf("b[]= %s ₩n ₩n");
18:  }
19:  void swap( char *p1, char *p2)
20:  {
21:     ....
22:  }
```

■ 실행화면

실행결과

```
<교환 전>
a[ ] = SEOUL
b[ ] = KOREA

<교환 후>
a[ ] = KOREA
b[ ] = SEOUL
```

C 언어 교육자료 : 문제			
분야	C Programming	제목	4.2 포인터 : 성적처리
수준	기본		
작성일			

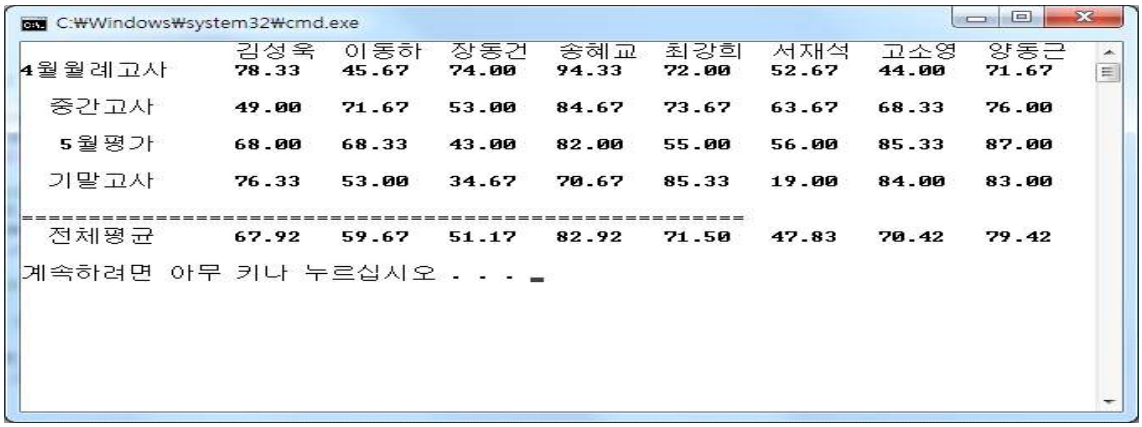
■ 문제 설명

◆ 학급의 학년도 평균 작성 : 아래 프로그램 템플릿 사용 (상수선언, 배열명, 함수명은 그대로 사용)

■ 프로그램 구조

```
1:  // 프로그램 헤더 : #include 문 등
2:
3:  #define EXAM_NUM      4  // 4월월례고사, 중간고사, 5월평가, 기말고사
4:  #define STUDENT_NUM   8 // 김성욱, 이동하 , ...,
5:  #define CLASS_NUM     3  // 국어 영어 수학
6:
7:  char StudentName[STUDENT_NUM][7]={"김성욱", "이동하", ... }
8:  char ExamName[EXAM_NUM][20]={"4월월례고사", "중간고사", "5월평가", "기말고사" }
9:
10:  int Score[EXAM_NUM][STUDENT_NUM][CLASS_NUM] =
11:    { {{65, 74, 96}, ... , {69, 70, 23} }, // 4월평가 : 김성욱(국영수), 이동하(국영수) ...
12:                  ...                          ...
13:      {{30, 66, 70}, ... , {30, 90, 93} } }; //기말고사 : 김성욱(국영수), 이동하(국영수) ...
14:
15:  int Avr_score[EXAM_NUM][STUDENT_NUM];       // 각 시험별 개인의 평균점수
16:  int Total_Avr_score[STUDENT_NUM];         // 전체시험에 대한 개인의 평균점수
17:
18:  main()
19:  {
20:  // 각 시험별 개인의 평균점수를 계산하는 코드
21:         - 계산된 결과값은 Avr_score[]에 입력됨, calculate_average_student()를 사용
22:         for (i=0; i < EXAM_NUM; i++)        // 시험별
23:            for (j=0; j < STUDENT_NUM; j++)      // 학생별
24:                Avr_score[][] = calculate_average_student()
25:
26:  // 앞에서 구한 점수를 활용 전체시험에 대한 개인의 평균점수
27:     for (i=0; i < STUDENT_NUM; i++)     // 학생별
28:            for (j=0; j < EXAM_NUM; j++)    // 시험별
29:        Total_Avr_score[] ← Avr_score[][]를 사용하여 계산
30:
31:  // 학생별로 평균점수를 화면에 디스플레이하는 프로그램
32:     for (i=0; i < STUDENT_NUM; i++)     // 학생별
33:     printf() 문.. // StudentName[], ClassName[] 사용하여 학생성명, 과목명 표기
34:  }
35:
36:  int calculate_average_student(int *p, int class_num)
37:  {
38:  // 학생의 과목별 점수 배열의 시작주소와 과목개수를 입력으로 받아 평균을 내서 출력하는 함수
39:  }
```

▶ 화면 출력 예시 (아래와 유사한 형식으로 - 조금 달라도 되지만)



C 언어 교육자료 : 문제			
분야	C Programming	제목	4.3 포인터 : 행렬곱
수준	기본		
작성일			

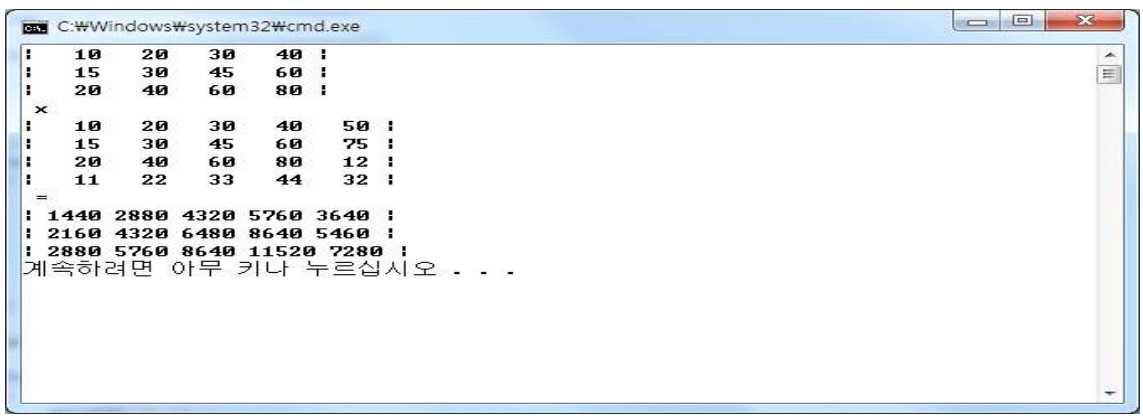
■ 문제 설명

◆ 행렬곱 : 아래 프로그램 템플릿 사용 (상수선언, 배열명, 함수명은 그대로 사용)

■ 프로그램 구조

```
23:  // 프로그램 헤더 : #include 문 등
24:
25:  #define MA_ROW    3 // 행렬 A의 행(Row) 개수
26:  #define MA_COLUMN                4     // 행렬 A의 열(Row) 개수
27:
28:  #define MB_ROW    4 // 행렬 B의 행(Row) 개수
29:  #define MB_COLUMN                5     // 행렬 B의 열(Row) 개수
30:
31:  int MatrixA[MA_ROW][MA_COLUMN] =
32:    { {10, 20, 30, 40},
33:        ....
34:      {40, 50, 60, 70}
35:    };
36:
37:  int MatrixB[MB_ROW][MB_COLUMN] =
38:    { {10, 20, 30, 40, 50},
39:        ....
40:      {50, 60, 70, 80, 90}
41:     };
42:
43:   int MatrixC[MA_ROW][MB_COLUMN];  // 행렬 곱(AxB)의 결과가 저장될 행렬
44:
45:  main()
46:  {
47:  * 행렬 곱을 계산하는 코드
48:       for (i=0; i < MA_ROW; i++)          // A의 행별
49:           for (j=0; j < MB_COLUMN; j++)     // B의 열별
50:              MatrixA[][] = calculate_vector_product()
51:
52:  * 결과를 디스플레이 하는 코드
53:         - 출력형식은 아래 <화면 출력 예시> 참조
54:  }
55:
56:  int calculate_vector_product(int *p1, int *p2, int num)
57:  {
58:  // - num 크기를 갖는 두개의 Vector(배열) 포인터(시작주소)를 입력으로 받아
59:  //   해당 벡터곱을 계산, 출력 ( FOR loop를 사용하면 됨)
60:  // - p2가 가리키는 것은 선형이 아닌 MatricB의 ROW임을 주의
61:  }
```

■ 화면 출력 예시 (아래 형식 : 줄은 정확하게 맞출것, 아래 예시의 값은 정확하지 않음 유의)



C 언어 교육자료 : 문제			
분야	C Programming	제목	4.4 포인터 : Linear queue
수준	기본		
작성일			

4.4.1 Linear Queue : 구조체 멤버를 표현하는데 있어 포인터를 사용하는 버전

■ **프로그램 구조**

```
1:  // 프로그램 헤더 : #include 문 등
2:  typedef struct score
3:   char   name[10];  // 성명
4:   char course[8];  // 과목명
5:   int   score;    // 점수
6:  } _SCORE;        // "_SCORE"란 "struct score" 타입에 대한 Alias로 "struct score"와 동일
7:
8:  #define QUEUE_SIZE  5  // QUEUE의 항목 크기
9:
10:  _SCORE LQueue[QUEUE_SIZE];  // Linear Queue로 항목개수가 QUEUE_SIZE 임
11:  _SCORE *Ptr;      // Queue의 항목에 대한 포인터로 사용
12:  main()
13:  {
14:    * 내부 변수 선언부 : 임시 포인터, 입력 등에 사용할 변수 등 선언
15:    * QUEUE 초기화
16:      - "Ptr"을 QUEUE의 시작주소로 셋팅
17:
18:    * 데이터를 1건 입력받아 QUEUE에 저장 (아래 라인1 ~ 3 참조)
19:        - 3개의 멤버 데이터를 화면에서 입력받아 큐에 등록
20:        - 큐 등록 등에는 반드시 "Ptr" 를 사용할 것 (즉 Ptr->name 과 같은 형식만 사용)
21:    * QUEUE 내용 디스플레이 및 계속할지 여부 처리 (아래 라인 8 참조)
22:        - 라인 8에서 "Y"를 쳐서 계속 입력하길 원하나 QUEUE가 꽉찬 경우에는
23:        QUEUE가 FULL이라는 메시지를 디스플레이해서 사용자에게 알림
24:  }
```

■ **화면 출력 예시**

◆ (QUEUE FULL의 경우는 나타나 있지 않으며, 개인적으로 알아서 표시)

```
1. 이름을 입력하시요     :  홍길동
2. 과목명을 입력하시요   :  영어
3. 점수를  입력하시요    :  80
4.
5. ** 현재까지 등록된 데이터는 1 건 입니다.
6.   1. 홍길동  영어  80
7.
8. 계속 입력하시겠읍니까 (Y/N) ?  Y
9.
10. 이름을 입력하시요    :  이기동
11. 과목명을 입력하시요  :  수학
12. 점수를  입력하시요   :  70
13.
14. ** 현재까지 등록된 데이터는 2 건 입니다.
15.   1. 홍길동  영어  80
16.   2. 이기동  수학  70
17.
18. 계속 입력하시겠읍니까 (Y/N) ?  N
19.
20. ** 큐내용 DISPLAY (2건)
21. 이름      홍길동   이기동
22. 과목명    영어     수학
23. 점수      80       70
24.
```

25. ** 프로그램이 종료하였읍니다.

※ 위에서 적색글자는 사용자가 입력해야 하는 부분임

4.4.2 Linear Queue : 구조체 멤버를 표현하는데 있어 구조체 변수를 사용하는 버전)

▶ 문제 4.4.1과 동일한 기능을 가지나 큐 등록 등에는 LQueue[3].name 과 같은 형식만 사용하여 구현한 버전

C 언어 교육자료 : 문제			
분야	C Programming	제목	5. 파일
수준	기본		
작성일			

5.1 파일 생성 및 활용

■ **문제 설명**

- File 생성 및 Input/Out
- 아래 프로그램을 완전한 프로그램으로 작성하시오

■ **프로그램 구조**

```
62:  // 프로그램 헤더 : #include 문 등
63:
64:  main()
65:  {
66:    * 파일 생성 및 쓰기
67:      - 파일생성 <파일명 : sample1>, <모드 : 쓰기 모드>
68:      - fputs()을 사용하여 "대한민국", "대구", "만촌동", 등과 같은 문자열을 여러개 기록
69:          - 파일 닫기
70:
71:    * 파일 일기
72:      - 위 파일열기 <모드 : 읽기 모드>
73:          - 파일 읽어 화면에 인쇄
74:          while () {
75:            // fgets() 등을 사용하여 1줄 읽어 화면에 인쇄
76:            // 파일 끝인가 검사하고 루프틀 빠져나와야 함
77:              }
78:      - 파일 닫기기록
79:  }
```