

[LN8] 합 수

1. 원의 반지름을 매개변수로 하는 함수 `double CircleArea(double radius)`를 이용하여 사용자가 입력하는 반지름 `r`에 대해 원의 넓이를 `float`형으로 출력하는 프로그램을 작성하시오.

[Sorce Code]

```
#include <stdio.h>

double CircleArea(double radius) {
    return 3.141592 * radius * radius; // 원넓이 구하는 공식
}

int main() {
    float a;

    printf("원의 반지름을 입력하시오:");
    scanf("%f", &a);

    printf("원의 면적은 %lf입니다.\n", CircleArea(a));

    return 0;
}
```

[Result]

```
원의 반지름을 입력하시오:5.0
원의 면적은 78.539800입니다.
```

2. 0 또는 1 값을 무작위로 한 번 생성하고, 이 함수의 호출 시마다 누적된 0의 생성된 횟수를 반환하는 함수 `b_rand()`를 이용하여 동전을 500번 던져 나온 앞면과 뒷면의 횟수를 각각 출력하는 프로그램을 작성하시오. (단, 함수 `b_rand()`는 정적 지역변수를 사용)

[Sorce Code]

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define THROW_NUM 1000
#define COIN_NUM 2

static int b_rand(int i);

int main() {
    int i, j;
    int fcnt = 0;
    int bcnt = 0;

    srand((unsigned)time(NULL));

    for (i = 0; i < THROW_NUM; i++)
        fcnt = b_rand(fcnt);

    bcnt = THROW_NUM - fcnt;

    printf("앞면이 나온 횟수: %d, 뒷면이 나온 횟수: %d\n", fcnt, bcnt);

    return 0;
}

static int b_rand(int i) {
    int coin;

    coin = rand() % COIN_NUM;
```

```
if(coin == 0)
    i++;

return i;
}
```

[Result]

앞면이 나온 횟수: 249, 뒷면이 나온 횟수: 251

3. 정수의 1차원 배열(원소의 수는 6)을 인자로 받아, 그 배열에 있는 원소들의 합을 구하는 함수 `int sum(int a[], int size)`와 그 배열의 각 원소의 값을 2배로 만드는 함수 `void twomul(int a[], int size)`를 이용하여 main 함수에서 `sum()` 및 `twomul()` 호출한 후에 배열의 값을 출력하는 프로그램을 작성하시오. 이때, 1차원 배열은 `int A[N_DATA] = {1, 2, 3, 4, 5, 6};`로 선언하시오.

[Source Code]

```
#include <stdio.h>

#define N_DATA 6

int sum(int a[], int size) {
    int sum = 0;

    for (int i = 0; i < size; i++)
        sum += a[i];

    return sum;
}

void twomul(int a[], int size) {
    for (int i = 0; i < size; i++)
        a[i] = 2 * a[i];
}

void printarr(int a[], int size) {
    for (int i = 0; i < size; i++)
        printf("%d ", a[i]);

    printf("\n");
}

int main() {
    int A[N_DATA] = { 1, 2, 3, 4, 5, 6 };

    printf("A배열의 값: ");
    printarr(A, N_DATA);

    printf("A배열의 합: %d\n", sum(A, N_DATA));
    twomul(A, N_DATA);

    printf("A배열을 2배로 한 후: ");
    printarr(A, N_DATA);

    return 0;
}
```

[Result]

A배열의 값: 1 2 3 4 5 6
A배열의 합: 21
A배열을 2배로 한 후: 2 4 6 8 10 12

[LN9] 배 열

1. 다음 [보기]에서 선언된 배열을 보고 최댓값과 최솟값을 출력하는 프로그램을 작성하시오.

[보기]

```
int arr[10] = { 60, 70, 80, 90, 75, 85, 95, 50, 45, 30 };
```

[Source Code]

```
#include <stdio.h>
#define SIZE 10

int main() {
    int arr[SIZE] = { 60, 70, 80, 90, 75, 85, 95, 50, 45, 30 };
    int max, min;
    int i;

    max = arr[0];
    min = arr[0];

    for (i = 0; i < SIZE; i++) {
        if (arr[i] > max)
            max = arr[i];
        if (arr[i] < min)
            min = arr[i];
    }

    printf("최댓값 = %d\n", max);
    printf("최솟값 = %d\n", min);

    return 0;
}
```

[Result]

```
최댓값 = 95
최솟값 = 30
```

2. 다음 4가지 조건을 모두 만족하는 극장 예약 프로그램을 작성하시오.

[조건 1] 배열을 이용
 [조건 2] 총 좌석 수는 10개
 [조건 3] 먼저 좌석 배치표를 보여줌
 [조건 4] 예약이 끝난 좌석은 1로, 예약이 안 된 좌석은 0으로 나타냄

[Source Code]

```
#include <stdio.h>
#define SIZE 10

int main() {
    char ans1;
    int ans2, i;
    int seats[SIZE] = {0};

    while(1) {
        printf("좌석을 예약하시겠습니까?(y또는n)");
        scanf(" %c", &ans1);

        if(ans1 == 'y') {
            printf("-----\n");
            printf(" 1 2 3 4 5 6 7 8 9 10\n");
            printf("-----\n");

            for(i = 0; i < SIZE; i++)
                printf(" %d", seats[i]);

            printf("\n몇 번째 좌석을 예약하시겠습니까?");
            scanf("%d", &ans2);

            if(ans2 <= 0 || ans2 > SIZE) {
                printf("1부터 10사이의 숫자를 입력하세요\n");
                continue;
            }

            if(seats[ans2-1] == 0) {
                seats[ans2-1] = 1;
                printf("예약되었습니다.\n");
            }
            else
                printf("이미 예약된 자리입니다.\n");
        }
        else if(ans1 == 'n')
            return 0;
    }

    return 0;
}
```

[Result]

좌석을 예약하시겠습니까?(y또는n)y

 1 2 3 4 5 6 7 8 9 10

0 0 0 0 0 0 0 0 0 0

몇 번째 좌석을 예약하시겠습니까?3
 예약되었습니다.

좌석을 예약하시겠습니까?(y또는n)y

 1 2 3 4 5 6 7 8 9 10

0 0 1 0 0 0 0 0 0 0

몇 번째 좌석을 예약하시겠습니까?7
 예약되었습니다.

좌석을 예약하시겠습니까?(y또는n)y

 1 2 3 4 5 6 7 8 9 10

0 0 1 0 0 0 1 0 0 0

몇 번째 좌석을 예약하시겠습니까?10
 예약되었습니다.

좌석을 예약하시겠습니까?(y또는n)n

3. 다음 6가지 조건을 만족하도록 성적처리 프로그램을 작성하시오.

배열 초기화를 이용하여 30명의 학생의 이름, 학번, 중간고사 및 기말고사 성적을 입력받아 각 학생별 최종성적을 기준에 의거하여 산출한 후, 학생의 석차를 계산하고 최종성적순으로 정렬하는 성적처리 프로그램을 작성하시오. (기준점수는 각 학생의 중간고사와 기말고사 성적의 합이며, 최종성적이 같은 경우는 이름순으로 출력)

최종성적기준	이름, 중간고사, 기말고사 데이터									
최상위 10%는 A+	강덕호	60	32	김준형	40	26	김재박	28	28	
	고기태	11	35	김민	66	59	임기현	30	31	
	다음 10%는 A	곽호철	36	19	박태수	41	41	이수복	45	90
	다음 20%는 B+	이도현	42	15	박병채	40	21	이원호	59	59
	다음 20%는 B	박동균	31	32	민현우	39	45	석주명	53	50
다음 10%는 C+	최동민	72	50	서인수	68	63	이범석	34	36	
다음 10%는 C	한명수	19	13	신인규	70	82	전호진	64	40	
다음 5%는 D+	김승호	50	32	성재훈	52	69	정광민	41	33	
다음 5%는 D	김은미	44	20	심동현	65	84	조영민	63	45	
다음 10%는 F	김태성	23	19	윤승한	31	25	홍정모	59	69	

[Source Code]

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAX 30

int main() {
    char name[MAX][10] = { // 학생의 이름 데이터 입력(배열 초기화 이용)
        "강덕호", "고기태", "곽호철", "이도현", "박동균",
        "최동민", "한명수", "김승호", "김은미", "김태성",
        "김준형", "김민철", "박태수", "박병채", "민현우",
        "서인수", "신인규", "성재훈", "심동현", "윤승한",
        "김재박", "임기현", "이수복", "이원호", "석주명",
        "이범석", "전호진", "전광민", "조영민", "홍정모"};

    int score[MAX][5] = { // 시험성적 데이터 입력(배열 초기화 이용)
        { 60, 32 }, { 11, 35 }, { 36, 19 }, { 42, 15 }, { 31, 32 },
        { 72, 50 }, { 19, 13 }, { 50, 32 }, { 44, 20 }, { 23, 19 },
        { 40, 26 }, { 66, 59 }, { 41, 41 }, { 40, 21 }, { 39, 45 },
        { 68, 63 }, { 70, 82 }, { 52, 69 }, { 65, 84 }, { 31, 25 },
        { 28, 28 }, { 30, 31 }, { 45, 90 }, { 59, 59 }, { 53, 50 },
        { 34, 36 }, { 64, 40 }, { 41, 33 }, { 63, 45 }, { 59, 69 } };

    int i, j; // 반복문 사용을 위한 변수 선언
    int rep, temp; // 버블 정렬을 위한 변수 선언
    int rank[MAX]; // 석차 배열 선언
    int s_temp[MAX]; // 최종성적 버블 정렬을 위한 임의의 배열 선언
    char n_temp[MAX]; // 학생이름 버블 정렬을 위한 임의의 배열 선언
    char grade[MAX][3];
    char swap; // 플래그 변수

    for (int i = 0; i < MAX; i++) // 최종성적 계산
        score[i][2] = (score[i][0] + score[i][1]) / 2;
    // 이름, 최종성적, 최종석차를 최종성적순으로 정렬
    for (rep = 1; rep < MAX; rep++) {
        swap = 'N'; // 플래그 변수의 초기화
        for (i = 0; i < MAX - rep; i++) {
            if (score[i][2] < score[i + 1][2]) {
                for (j = 0; j < 4; j++) {
                    s_temp[i] = score[i][j];
                    score[i][j] = score[i + 1][j];
                    score[i + 1][j] = s_temp[i];
                }
            }
        }
    }
}
```

```

    strcpy(n_temp, name[i]);
    strcpy(name[i], name[i + 1]);
    strcpy(name[i + 1], n_temp);
    swap = 'Y';
}
// 최종성적이 같은 경우 이름의 '가나다' 순으로 출력
else if (score[i][2] == score[i + 1][2] && strcmp(name[i], name[i + 1]) > 0) {
    strcpy(n_temp, name[i]);
    strcpy(name[i], name[i + 1]);
    strcpy(name[i + 1], n_temp);
    swap = 'Y';
}
}
if (swap == 'N')
    break;
// 학생의 석차 계산
for (i = 0; i < MAX; i++) {
    rank[i] = 1;
    for (j = 0; j < MAX; j++) {
        if (score[i][2] < score[j][2])
            rank[i]++;
    }
}
// 학생의 최종성적을 상대평가로 산출
for (int i = 0; i < MAX; i++) {
    if (rank[i] <= MAX * 0.1) // 최상위 10%
        strcpy(grade[i], "A+");
    else if (rank[i] <= MAX * 0.2 && rank[i] > MAX * 0.1) // 다음 10%
        strcpy(grade[i], "A");
    else if (rank[i] <= MAX * 0.4 && rank[i] > MAX * 0.2) // 다음 20%
        strcpy(grade[i], "B+");
    else if (rank[i] <= MAX * 0.6 && rank[i] > MAX * 0.4) // 다음 20%
        strcpy(grade[i], "B");
    else if (rank[i] <= MAX * 0.7 && rank[i] > MAX * 0.6) // 다음 10%
        strcpy(grade[i], "C+");
    else if (rank[i] <= MAX * 0.8 && rank[i] > MAX * 0.7) // 다음 10%
        strcpy(grade[i], "C");
    else if (rank[i] <= MAX * 0.85 && rank[i] > MAX * 0.8) // 다음 5%
        strcpy(grade[i], "D+");
    else if (rank[i] <= MAX * 0.9 && rank[i] > MAX * 0.85) // 다음 5%
        strcpy(grade[i], "D");
    else // 다음 10%
        strcpy(grade[i], "F");
}
// 성적처리 프로그램 최종출력
printf("\n");
printf("성적처리 프로그램\n");
printf("\n");
printf("=====\n");
printf("이름  중간고사  기말고사  최종성적  최종석차\n");
printf("=====\n");

for (i = 0; i < MAX; i++)
    printf("    %s%7d점%9d점    %s    \t    %d\n", name[i], score[i][0], score[i][1], grade[i], rank[i]);
printf("=====\n");

return 0;
}

```

[Result]

성적처리 프로그램

이 름	중간고사	기말고사	최종성적	최종석차
신인규	70점	82점	A+	1
심동현	65점	84점	A+	2
이수복	45점	90점	A+	3
서인수	68점	63점	A	4
홍정모	59점	69점	A	5
김민철	66점	59점	A	6
최동민	72점	50점	B+	7
성재훈	52점	69점	B+	8
이원호	59점	59점	B+	9
조영민	63점	45점	B+	10
전호진	64점	40점	B+	11
석주명	53점	50점	B+	12
강덕호	60점	32점	B	13
민현우	39점	45점	B	14
김승호	50점	32점	B	15
박태수	41점	41점	B	15
전광민	41점	33점	B	17
이범석	34점	36점	B	18
김준형	40점	26점	C+	19
김은미	44점	20점	C+	20
박동균	31점	32점	C+	21
박병채	40점	21점	C	22
임기현	30점	31점	C	22
김재박	42점	15점	C	24
윤승한	31점	25점	C	24
이도현	28점	28점	C	24
곽호철	36점	19점	D	27
고기태	11점	35점	F	28
김태성	23점	19점	F	29
한명수	19점	13점	F	30

[LN10] 포인터

1. 두 점 (2, 5)와 (4, 9)를 지나는 일차함수 식을 구하는 프로그램을 작성하시오. (기울기와 y절편 출력)

[Source Code]

```
#include <stdio.h>

int get_line_parameter(int x1, int y1, int x2, int y2, float *slope, float *yintercept) {
    if( x1 == x2 )
        return -1;
    else {
        *slope = (float)(y2 - y1)/(float)(x2 - x1);
        *yintercept = y1 - (*slope) * x1;

        return 0;
    }
}

int main() {
    float s, y;

    if( get_line_parameter(2, 5, 4, 9, &s, &y) == -1 )
        printf("Error!!!\n");
    else
        printf("기울기 = %.2f\ny절편 = (0.00, %.2f)\n", s, y);

    return 0;
}
```

[Result]

```
기울기 = 2.00
y절편 = (0.00, 1.00)
```


2. 학생들의 평점은 4.5점이 만점이라고 하자. 배열 grades[]에 학생 10명의 학점이 저장되어 있다. 이것을 100점 만점으로 변환하여서 배열 scores[]에 저장하는 함수를 작성하고, 변환 결과를 출력하라.

학생들의 평점 만점은 4.5점으로 가정하고, 배열 grades[]에 학생 10명의 학점이 저장되어 있다. 이것을 100점 만점으로 변환하여서 배열 scores[]에 저장하는 함수를 작성하여 변환 결과를 출력하는 프로그램을 작성하라.

[보기]

```
double grades[10] = {1.5, 2, 3, 4.5, 3, 2.5, 1, 2, 3.5, 4}
void convert(double *grades, double *scores, int size);
```

[Source Code]

```
#include <stdio.h>

void convert(double *grades, double *scores, int size) {
    int i;
    for(i = 0; i < size; i++)
        scores[i] = 100.0 * (grades[i] / 4.5);
}

void printarr(double *a, int size) {
    int i;
    for(i = 0; i < size; i++)
        printf("%.1f\n", a[i]);
    printf("\n");
}

int main() {
    double grades[10] = {1.5, 2, 3, 4.5, 3, 2.5, 1, 2, 3.5, 4};
    double scores[10] = {0};

    printf("<grades의 값> \n");
    printarr(grades, 10);

    convert(grades, scores, 10);

    printf("<scores의 값> \n");
    printarr(scores, 10);

    return 0;
}
```

[Result]

<grades의 값>

1.5
2.0
3.0
4.5
3.0
2.5
1.0
2.0
3.5
4.0

<scores의 값>

33.3
44.4
66.7
100.0
66.7
55.6
22.2
44.4
77.8
88.9

3. 학생들의 성적이 scores라는 2차원 배열에 저장되어 있다고 가정하고, scores의 행은 한 학생에 대한 여러 번의 시험 성적을 나타내며, scores의 열은 한 시험에 대한 여러 학생들의 성적이다. 학생별로 성적의 평균을 2차원 배열의 각 행이 1차원 배열임을 이용하여 1차원 배열의 평균을 구하는 함수 `get_average(int *list, int n)`를 호출하여 각 학생에 대한 평균성적을 계산하는 프로그램을 작성하시오.

[보기]

	시험1	시험2	시험3
학생1	100	30	67
학생2	89	50	12
학생3	19	60	90

[Sorce Code]

```
#include <stdio.h>

int get_average(int *list, int n);

int main() {
    int score[3][3] = { { 100, 30, 67 }, { 89, 50, 12 }, { 19, 60, 90 } };

    for (int i = 0; i < 3; i++)
        printf("학생%d의 시험 평균은 %d 입니다.\n", i + 1, get_average(*score, i));

    return 0;
}

int get_average(int *list, int n) {
    int ave = 0;
    int j = 0;

    for (n; j < 3; n++) {
        ave += *(list + n);
        j++;
    }

    ave /= 3;

    return ave;
}
```

[Result]

학생1의 시험 평균은 65 입니다.
 학생2의 시험 평균은 50 입니다.
 학생3의 시험 평균은 56 입니다.

[LN11] 문자열 처리

1. 1차원 배열에 문자를 입력받아 역순으로 출력되도록 프로그램을 작성하시오.

[Source Code]

```
#include <stdio.h>
#include <string.h>

int main() {
    char name[20] = "Hello World!";
    int i;

    for(i = strlen(name) - 1; i >= 0; i--)
        printf("%c", name[i]);
    printf("\n");

    return 0;
}
```

[Result]

!dlrow olleH

2. 두 개의 서로 다른 문자열을 입력받아 하나의 문자열로 합치는 프로그램을 작성하세요. (strcat(), strlen()를 사용)

[Source Code]

```
#include <stdio.h>
#include <string.h>

void RemoveBSN(char str[]) {
    int len = strlen(str);
    str[len - 1] = 0;
}

int main() {
    char str[50];
    char ch[50];

    printf("첫번째 문자열 입력 : ");
    fgets(str, sizeof(str), stdin);

    RemoveBSN(str);

    printf("두번째 문자열 입력 : ");
    fgets(ch, sizeof(ch), stdin);

    strcat(str, ch);
    puts(str);

    return 0;
}
```

[Result]

첫번째 문자열 입력 : Hello!
두번째 문자열 입력 : My name is Changyeon Jo
Hello! My name is Changyeon Jo

3. 사용자가 입력한 영문자, 숫자, 구두점(. ! 3가지)의 개수를 세는 프로그램을 작성하시오. (사용자가 입력한 영문자, 숫자, 구두점은 if문을 사용하여 분류하고, <Enter> 키를 누르면 입력이 중단되어 프로그램이 종료된다. 이때 각각의 개수를 출력)

[Sorce Code]

```
#include <stdio.h>
#include <conio.h> //_kbhit()함수와 _getch()함수를 포함하고 있는 헤더파일
#define ENTER 0x0d //ENTER값을 0x0d로 지정 (이때, 0x0d는 숫자 13의 ASCII 코드이다.)

int main() {
    int letter = 0; // letter를 정수형으로 선언한 뒤 0으로 초기화
    int num = 0; // num을 정수형으로 선언한 뒤 0으로 초기화
    int mark = 0; // mark를 정수형으로 선언한 뒤 0으로 초기화

    //c를 문자형으로 선언한 후, 0으로 초기화시킨 후
    for(char c=0;;) { //아래 조건들의 모든 작업이 마무리될 때까지 무한 반복
        if (kbhit()) { //키보드의 키값을 입력 받았다면
            c = getch(); //입력된 키값을 저장한다.
            if ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z')) //영문자 조건
                letter++; //영문자의 개수가 1씩 증가한다.
            else if (c >= 0x30 && c <= 0x39) //숫자 조건
                num++; //숫자의 개수가 1씩 증가한다.
            else if (c == ',' || c == '.' || c == '!') //구두점 조건
                mark++; //구두점의 개수가 1씩 증가한다.
            else if (c == ENTER) //<ENTER>키를 누르면
                break; //종료한다
        }
    }

    printf("영문자의 개수 : %d\n", letter); //영문자 개수를 출력한다.
    printf("숫자의 개수 : %d\n", num); //숫자의 개수를 출력한다.
    printf("구두점의 개수 : %d\n", mark); //구두점의 개수를 출력한다.

    return 0;
}
```

[Result]

영문자의 개수 : 58
숫자의 개수 : 27
구두점의 개수 : 26