

1.

```
int main()
{
    srand(time(NULL));
    double score[STUDENT][CATEGORY] = { 0 };
    double total_score[STUDENT] = { 0 };
    double max = -100;
    double min = 100;

    for (int i = 0; i < STUDENT; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            score[i][j] = rand() % 100 + 1;
        }

        score[i][4] = rand() % 10 + 1;

        printf("%2d 번 학생의 중간 점수: %.0f\n", i + 1, score[i][0]);
        printf("        기말 점수: %.0f\n", score[i][1]);
        printf("        기말 과제: %.0f\n", score[i][2]);
        printf("        발표 점수: %.0f\n", score[i][3]);
        printf("        결석 점수: %.0f\n", score[i][4]);
    }
    printf("\n");

    for (int i = 0; i < STUDENT; i++)
    {
        total_score[i] = (score[i][0] * 0.3) + (score[i][1] * 0.4) + (score[i][2] * 0.2) + (score[i][3] * 0.1) - (score[i][4] * 1);
        printf("%2d 번 학생의 총 점수: %.2f\n", i + 1, total_score[i]);
    }
    printf("\n");

    for (int i = 0; i < STUDENT; i++)
    {
        if (max < total_score[i])
        {
            max = total_score[i];
        }

        if (min > total_score[i])
        {
            min = total_score[i];
        }
    }

    printf("최종성적 1 등: %.2f\n", max);
    printf("최종성적 20 등: %.2f\n", min);

    return 0;
}
```

```
1 번 학생의 총 점수: 50.50
2 번 학생의 총 점수: 59.10
3 번 학생의 총 점수: 76.40
4 번 학생의 총 점수: 37.70
5 번 학생의 총 점수: 60.20
6 번 학생의 총 점수: 31.90
7 번 학생의 총 점수: 38.70
8 번 학생의 총 점수: 23.40
9 번 학생의 총 점수: 36.70
10 번 학생의 총 점수: 62.00
11 번 학생의 총 점수: 62.30
12 번 학생의 총 점수: 39.80
13 번 학생의 총 점수: 54.50
14 번 학생의 총 점수: 61.10
15 번 학생의 총 점수: 51.40
16 번 학생의 총 점수: 50.70
17 번 학생의 총 점수: 59.80
18 번 학생의 총 점수: 50.10
19 번 학생의 총 점수: 16.70
20 번 학생의 총 점수: 44.50
```

```
최종성적 1 등: 76.40
최종성적 20 등: 16.70
```

2.

```
int main() {
    int evenArray[10];
    int oddArray[10];
    int evenCount = 0;
    int oddCount = 0;

    srand(time(NULL));

    while (evenCount < 10 && oddCount < 10)
    {
        int randomNumber = rand() % 100;

        if (randomNumber % 2 == 0)
        {
            evenArray[evenCount] = randomNumber;
            evenCount++;
        }
        else
        {
            oddArray[oddCount] = randomNumber;
            oddCount++;
        }
    }

    printf("짝수 배열: ");
    for (int i = 0; i < evenCount; i++)
    {
        printf("%d ", evenArray[i]);
    }
    printf("\n");

    printf("홀수 배열: ");
    for (int i = 0; i < oddCount; i++)
    {
        printf("%d ", oddArray[i]);
    }
    printf("\n");

    printf("짝수 배열 데이터 개수: %d\n", evenCount);
    printf("홀수 배열 데이터 개수: %d\n", oddCount);

    return 0;
}
```

```
짝수 배열: 0 58 94 56 6 76 96 2
홀수 배열: 85 47 1 17 93 87 71 33 57 87
짝수 배열 데이터 개수: 8
홀수 배열 데이터 개수: 10
```

3.

```
int main()

    srand(time(NULL));

    int size = 10;

    int num[10];

    for (int i = 0; i < size; i++)
    {
        num[i] = rand() % 100 + 1;
    }
    printf("배열 초기 상태: ");
    printf("%d ", num[0]);

    printf("\n");

    for (int i = 1; i < size; i++) {
        int key = num[i];
        int j = i - 1;

        printf("정렬 중인 원소 %d -> ", key);
        for (int k = 0; k <= j; k++) {
            printf("%d ", num[k]);
        }
        printf("\n");

        while (j >= 0 && num[j] > key) {
            num[j + 1] = num[j];
            j = j - 1;
        }

        num[j + 1] = key;

    }

    printf("배열 정렬 후: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", num[i]);
    }
    printf("\n");

    return 0;
}
```

```
배열 초기 상태: 56
정렬 중인 원소 71 -> 56
정렬 중인 원소 27 -> 56 71
정렬 중인 원소 96 -> 27 56 71
정렬 중인 원소 6 -> 27 56 71 96
정렬 중인 원소 12 -> 6 27 56 71 96
정렬 중인 원소 40 -> 6 12 27 56 71 96
정렬 중인 원소 100 -> 6 12 27 40 56 71 96
정렬 중인 원소 91 -> 6 12 27 40 56 71 96 100
정렬 중인 원소 51 -> 6 12 27 40 56 71 91 96 100
배열 정렬 후: 6 12 27 40 51 56 71 91 96 100
```