

# 자료구조

김태원

2023년 10월 1일

## 제1장

### C 리뷰

1. 입력으로 하나의 양의 정수를 받은 후 0이 될 때까지 연속적으로 2로 나눈 몫을 출력하는 프로그램을 작성하라.

```
1 #include <stdio.h>
2 int main(){
3     int n;
4     scanf("%d", &n);
5     for(int i=n; i!=0; i/=2)
6         printf("%d ", i);
7 }
```

쉬운 문제다. `i/=2`는 그냥 멧내는 용이다.

2. 입력으로 하나의 양의 정수  $n$ 을 받은 후 다음의 합을 구하여 출력하는 프로그램을 작성하라. 단, 소수점 4자리까지만 출력하라.

$$1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$$

```
1 #include <stdio.h>
2 int main(){
3     int n;
4     double sum = 0;
5     for(double r=1; r<=n; r++)
6         sum = sum + 1/r;
7     printf("%.4f\n", sum);
8 }
```

5번 라인의 for문에서 `r`을 `double`로 선언했다. `int`로 선언하면 모든 입력에 대해 출력은 1이기 때문이다.

3. 입력으로 하나의 양의 정수  $n$ 을 받은 후 다음의 합을 구하여 출력하는 프로그램을 작성하라.

단, 소수점 4자리까지만 출력하라.

$$1 + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!}$$

```
1 #include <stdio.h>
2 int main(){
3     int n;
4     double sum = 0;
5     scanf("%d", &n);
6     for(double i=1; i<=n; i++){
7         double frac = 1;
8         for(double j=i; j>0; j--){
9             frac *= j;
10            sum += 1/frac;
11        }
12        printf("%.4f\n", sum);
13    }
```

순서대로 해결했다. 9번 라인이 아니라 7번 라인에서 `frac`을 선언한 이유는 10번 라인에서 `frac`이 쓰이기 때문이다.

4. 먼저 입력될 정수의 개수  $n \leq 100$ 을 입력받고, 이어서  $n$ 개의 정수를 받아 평균과 표준편차를 계산하여 소수점 이하 4번째 자리까지 출력하는 프로그램을 작성하라. 표준편차는 다음과 같이 정의된다. 루트를 계산하기 위해서 `math.h`를 `include`하고 `sqrt`함수를 사용하라.

$$SD = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 int main(){
5     int n;
6     int *arr;
7     double avg = 0;
8     double sum = 0;
9     scanf("%d", &n);
10    arr = (int *)malloc(sizeof(int)*n);
11    for(int i=0; i<n; i++){
12        int p;
13        scanf("%d", &p);
```

```

14     arr[i] = p;
15     avg += arr[i];
16 }
17 for(int i=0; i<n; i++)
18     sum += pow(arr[i]-avg/n, 2.0);
19 printf("%.5g\n", sqrt(sum/n));
20 free(arr);
21 }

```

`.5g` 라는 format specifier를 사용했는데, 가령 `4.5000...`를 `4.5`로 잘라서 출력하기 위해서다. 참고로 `math.h` 라이브러리를 포함한 파일을 컴파일하려면 `-lm` 명령어를 덧붙여야 한다.

5. 먼저 입력될 정수의 개수  $2 \leq n \leq 100$ 을 입력받고, 이어서  $n$ 개의 정수를 입력받는다. 입력된 정수들 중에서 최소값과 두 번째로 작은 값을 찾아 출력하는 프로그램을 작성하라. 만약 최소값이 2개 이상 중복되어 존재하면 그 중 하나를 최소값으로, 다른 하나를 두 번째로 작은 값으로 간주한다.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(){
4     int n, key;
5     int *arr;
6     scanf("%d", &n);
7     arr = (int*)malloc(sizeof(int)*n);
8     for(int i=0; i<n; i++)
9         scanf("%d", &arr[i]);
10    for(int i=1; i<n; i++){
11        key = arr[i];
12        int j;
13        for(j=i-1; j>=0 && arr[j]>key; j--)
14            arr[j+1] = arr[j];
15        arr[j+1] = key;
16    }
17    printf("%d %d\n", arr[0], arr[1]);
18    free(arr);
19 }

```

10번 라인에서 시작되는 for 블록은 삽입정렬 알고리즘을 구현한 것이다. 삽입 정렬 알고리즘은 아

---

**Algorithm 1:** Insertion Sort

---

```

Input:  $A$ : Array
for  $i \in \{2, \dots, A.length\}$  do
     $key \leftarrow A[i]$ 
     $j \leftarrow i - 1$ 
    while  $j > 0 \ \& \ A[j] > key$  do
         $A[j + 1] \leftarrow A[j]$ 
         $j \leftarrow j - 1$ 
     $A[j + 1] \leftarrow key$ 

```

래와 같은 방식으로 작동한다.

$$\begin{aligned}
A &= [7, 3, 1, 2, 4, 6] && \text{첫 번째 for 루프: } 7 = A[1] > \text{key} = A[2] = 3 \\
&&& \mapsto [3, 7, 1, 2, 4, 6] \\
&&& \text{두 번째 for 루프: } 7 = A[2] > \text{key} = A[3] = 1 \\
&&& \mapsto [3, 1, 7, 2, 4, 6] \\
&&& \text{두 번째 for 루프: } 3 = A[1] > \text{key} = A[3] = 1 \\
&&& \mapsto [1, 3, 7, 2, 4, 6] \\
&&& \text{세 번째 for 루프: } 7 = A[3] > \text{key} = A[4] = 2 \\
&&& \mapsto [1, 3, 2, 7, 4, 6] \\
&&& \text{세 번째 for 루프: } 3 = A[2] > \text{key} = A[4] = 2 \\
&&& \mapsto [1, 2, 3, 7, 4, 6] \\
&&& \text{네 번째 for 루프: } 7 = A[4] > \text{key} = A[5] = 4 \\
&&& \mapsto [1, 2, 3, 4, 7, 6] \\
&&& \text{다섯 번째 for 루프: } 7 = A[5] > \text{key} = A[6] = 6 \\
&&& \mapsto [1, 2, 3, 4, 6, 7]
\end{aligned}$$