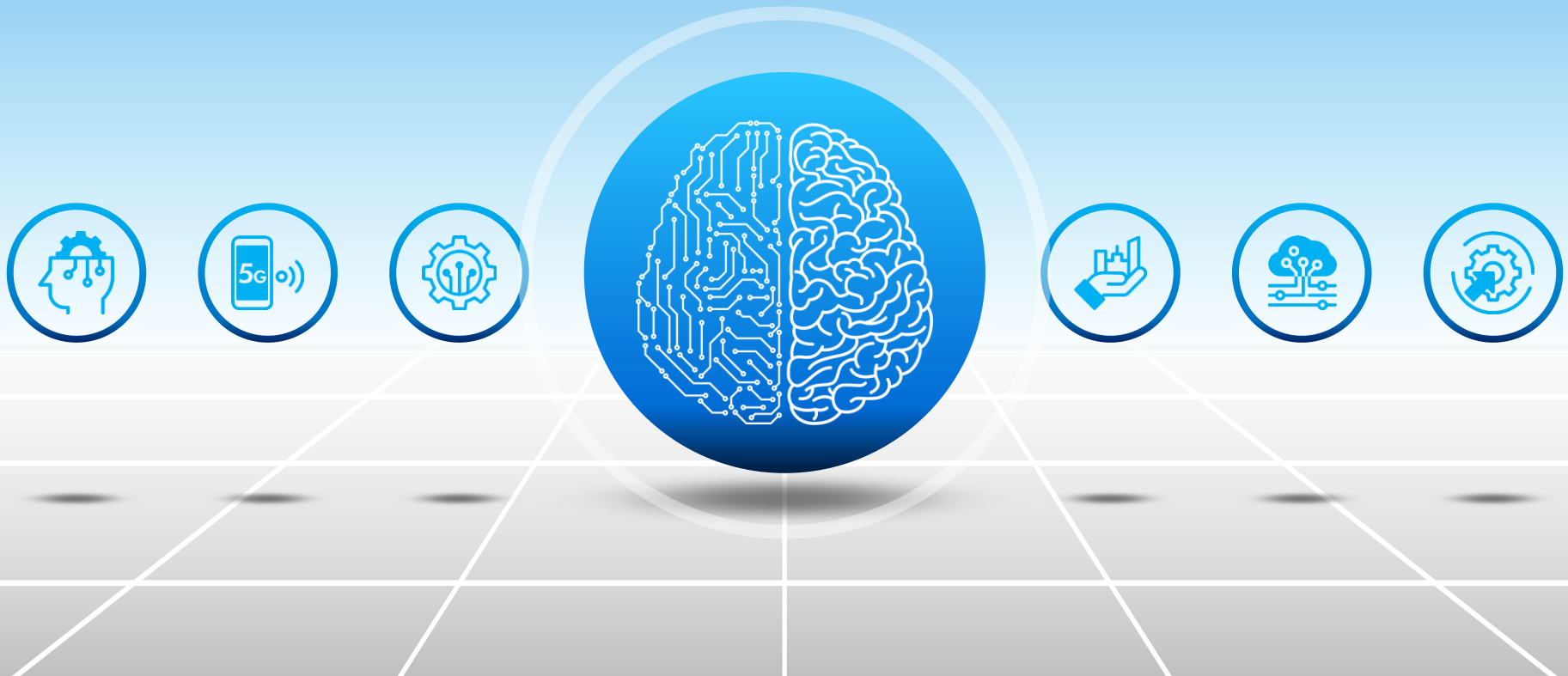


CIS3034 문제해결프로젝트

반복문을 활용한 문제해결



목 차

CONTENTS



I

반복문 활용 예제풀이

예제 풀이



문제1E-합 구하기2

사막 여우의 탈을 쓴 지수는 놀이공원에서 아르바이트를 하고 있습니다. 지수가 담당하고 있는 놀이기구는 무게에 민감하기 때문에 몇 가지 제한을 두고 있습니다.

- 이 놀이기구는 몸무게가 P kg이 넘는 사람은 탑승 할 수 없습니다.
- 그리고 탑승한 승객의 몸무게의 총 합은 Q kg을 넘을 수 없습니다.
- 다만 탑승하는 인원의 제한은 두고 있지 않습니다.

보통은 놀이기구의 최대 탑승 중량을 넘기 전에 탑승 인원을 나누기 때문에 크게 문제가 발생하지 않습니다. 하지만 오늘은 조금 문제가 있습니다. 아주대학교 알고리즘 동아리라고 자신들을 소개한 단체손님 때문입니다. 그들은 애초에 몸무게가 P kg를 넘어가는 회원은 제외하더라도 나머지 인원들은 꼭 함께 놀이기구를 탑승해야한다고 우기고 있습니다. 이 동아리의 회원들 중 놀이기구를 탈 수 있는 사람은 몇 명인지, 그리고 체중 제한을 통과한 사람들은 모두 함께 놀이기구를 탈 수 있는지 계산할 수 있는 프로그램을 작성해주세요.

입력 형식

첫 줄에 세 자연수 N, P, Q 가 주어진다. 각각 동아리 회원의 수 N , 놀이기구의 탑승 제한 체중 P , 놀이기구의 최대 하중 Q 이다.

그리고 두 번째 줄에는 각 회원의 몸무게가 주어진다.

- N 는 10,000이하의 자연수이다.
- P 와 각 동아리 회원의 몸무게는 1이상 300이하의 자연수이다.
- Q 는 3,000,000이하의 자연수이다

출력 형식

첫 줄에 놀이기구 체중 제한에 걸리지 않은 회원의 수와 그들의 몸무게의 총 합을 공백으로 구분하여 출력한다.

그리고 두 번째 줄에 그들 모두가 함께 놀이기구에 탑승할 수 있는지 여부를 YES 혹은 NO로 출력한다.

문제 1E-합 구하기2

입/출력 예시



: 공백



: 줄바꿈

예시 1

입력

10 80 500

75 50 67 84 54 67 98 85 58 90

출력

6 371

YES

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

using namespace std;

/**
 * 동아리 멤버들의 탑승 가능 여부를 출력하는 함수
 *
 * @param data 각 멤버들의 몸무게 data[0] ~ data[n-1]
 * @param n 멤버들의 수
 * @param p 놀이기구 탑승 가능 체중 제한
 * @param q 놀이기구 최대 하중
 */
void solve(int data[], int n, int p, int q)
{
}
```

```
int main()
{
    int n, p, q;
    int *data;

    scanf("%d %d %d", &n, &p, &q);
    data = new int[n];
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &data[i]);
    }

    solve(data, n, p, q);

    delete[] data;
    return 0;
}
```

문제 1F-탐색하기 1

N 개의 정수로 이루어진 배열과 찾고자 하는 값 M 이 주어진다. 이 배열에서 M 이 존재하는 인덱스를 출력하는 프로그램을 작성하시오.

입력 형식

첫 줄에는 두 자연수 N 과 M 이 주어진다. N 은 1만 이하의 자연수이다.

두 번째 줄에는 N 개의 서로 다른 자연수가 공백으로 구분되어 주어진다.

입력되는 모든 숫자와 M 은 1이상 1,000이하의 자연수임이 보장된다.

출력 형식

한 줄에 입력된 배열에서 M 이 존재하는 인덱스를 정수로 공백없이 출력한다. 인덱스는 0이상 $N-1$ 이하의 정수라고 가정한다. 찾고자 하는 값이 배열에 없다면 -1 을 출력한다.

입/출력 예시

 : 공백  : 줄바꿈

예시 1

```
입력
3 2
2 3 1

출력
0
```

예시 2

```
입력
10 4
10 7 2 6 3 1 88 27 35 85

출력
-1
```

문제 1F-탐색하기 1

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

using namespace std;

/**
 * 배열에서 특정 원소의 위치를 찾는 함수
 * @param data 중복 없는 정수 배열 data[0] ~ data[n-1]
 * @param n 배열의 크기 n
 * @param m 배열에서 찾고자 하는 원소
 * @return 원소가 존재한다면 인덱스를, 존재하지 않으면 -1을 반환한다.
 */
int findIndex(int data[], int n, int m)
{

}

int main()
{
    int n, m;
    int *data;

    scanf("%d %d", &n, &m);
    data = new int[n];
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &data[i]);
    }

    int answer = findIndex(data, n, m);

    printf("%d\n", answer);
    delete[] data;
    return 0;
}
```

문제 1G-탐색하기 2

민규와 주명이는 대한민국의 국민입니다. 국민으로서 부여받은 병역의 의무를 이행하기 위하여 둘은 동반 입대를 결정했습니다. 하지만 동반 입대를 단순히 동시에 입영 신청을 하는 것으로 착각한 둘은 같은 시각에 입영 신청을 하고 입대를 기다리고 있었습니다. 하지만 서로 다른 곳에서 복무를 하게 된다는 사실을 알게 된 둘은 경악했습니다. 또한 비슷한 시기에 입영 신청을 한 같은 과 친구들도 모두 뿔뿔이 흩어져 다른 부대로 가게 되었다는 사실을 알게 되었습니다. 결국 외로움을 참지 못한 주명이는 병무청을 해킹하여 복무지를 조작하기로 결정하였습니다. 주명이는 최근 복무지가 결정된 데이터 N 개를 처리된 시간 순서대로 얻을 수 있었습니다. 주명이는 같은 학교 친구들이 모두 같은 복무지에 갈 수 있도록 하고자 합니다. 하지만 일일이 데이터를 확인할 수 없던 주명이는 처음 처리된 아주대학교 학생부터 가장 마지막에 처리된 아주대학교 학생 까지의 정보만을 같은 복무지로 수정하고자 합니다.

주명이가 얻은 복무지 처리 데이터의 수 N 과 그 데이터들이 순서대로 주어질 때, 같은 복무지로 처리해야 할 첫 번째 데이터의 번호와 마지막 데이터의 번호를 구하는 프로그램을 작성해주세요.

입력 형식

첫 줄에는 주명이가 얻은 데이터의 수 N 이 10,000이하의 자연수로 주어진다.

두 번째 줄 부터는 총 N 줄에 걸쳐 각 줄에 하나씩 해당 학생의 소속 학교가 주어진다. 시간 순으로 첫 번째 데이터가 가장 먼저 주어진다.

소속 학교명은 공백이 없는 알파벳 대문자만으로 구성되어 있으며, 1글자 이상 10글자 이하이다.

소속 학교명이 **AJOU**인 모든 데이터는 아주대학교 소속 학생의 데이터라고 가정하며, 최소 한 명 이상의 학생은 아주대학교 소속이다.

출력 형식

주명이가 같은 복무지로 처리 할 첫 데이터와 마지막 데이터의 번호를 한 줄에 공백으로 구분하여 출력한다.

문제 1G-탐색하기2

입력

```
10
HELL
CHOSUN
SKK
KOREA
AJOU
YONSEI
SEOUL
AJOU
AJOU
WOOSONG
```

출력

```
5.9
```

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

using namespace std;

/**
 * 배열에서 소속이 "AJOU"인 첫 원소와 마지막 원소를 출력하는 함수
 * @param school: 각 사람들의 소속학교 정보 배열
 * @param n: .....사람들의 수
 */
void printIndexes(string school[], int n)
{
    int first = -1; // 존재하지 않으면 -1
    int last = -1; // 존재하지 않으면 -1

    printf("%d %d\n", first, last);
}
```

```
int main()
{
    int n;
    char buff[11];
    string* school;

    scanf("%d", &n);
    school = new string[n];

    for (int i = 0; i < n; i++)
    {
        scanf("%s", buff);
        school[i] = buff;
    }

    printIndexes(school, n);

    delete[] school;
    return 0;
}
```

문제 1H-탐색하기 3

데이터 마이닝 과목을 수강하게 된 지애는 평소 수학과 컴퓨터와는 거리가 먼 삶을 살아왔기 때문에 과제에 애를 먹고있다. 평소 지애 괴롭히기를 좋아하는 미주는 진지하게 휴학을 고민하는 지애가 학교를 떠나는 것을 원하지 않기에 과제를 대신 해주기로 했다. 하지만 미주도 프로그래밍을 전혀 할 줄 모르기 때문에 평소 컴퓨터를 잘 다루기로 소문난 선배인 당신에게 과제를 부탁했다. 당신은 미주를 위하여 프로그램을 작성해야 한다.

N 개의 수치 데이터가 정수로 주어진다. 당신은 이 개별 데이터들 중 전체 데이터의 평균과 가장 가까운 데이터와 그 번호를 출력하고자 한다. 가깝다라는 의미는 평균에서 그 숫자를 뺀 절댓값이 작다는 의미이다. 평균과의 거리가 같은 숫자가 두 개 이상일 경우, 가장 먼저 입력된 데이터를 우선으로 한다. 데이터의 번호는 입력받은 순서대로 1부터 N 으로 부여된다.

입력 형식

첫 줄에 전체 데이터의 수 N 이 10,000이하의 자연수로 주어진다.

두 번째 줄에는 각 수치 데이터가 공백으로 구분되어 한 줄에 주어진다. 각 수치 데이터는 -100,000 이상 100,000 이하의 정수이다.

출력 형식

한 줄에 평균과 가장 거리가 가까운 데이터의 번호와 그 값을 공백으로 구분하여 출력한다.

입/출력 예시

 : 공백  : 줄바꿈

예시 1

입력

6

1 2 4 7 10 15

출력

4 7

문제 1H-탐색하기3

```
#include<stdio>
#include<cmath>
#include<iostream>

using namespace std;

/**
 * 평균과의 차가 가장 작은 데이터의 번호를 반환하는 함수
 * 여러 가지라면 가장 빠른 번호를 반환한다.
 *
 * @param data
 * @param n
 * @return int 가장 평균과 가까운 데이터의 번호 (1번부터 시작)
 */
int findIndex(int data[], int n)
{
}
```

```
int main()
{
    int n;
    int* data;

    scanf("%d", &n);
    data = new int[n];

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &data[i]);
    }

    int answer = findIndex(data, n);
    printf("%d\n", answer);

    delete[] data;
    return 0;
}
```

문제 11-선택정렬 (Selection Sort)

배운 내용을 바탕으로 선택 정렬을 구현해봅시다. 선택 정렬은 아래와 같은 연산을 반복하여 구현할 수 있습니다.

1. 0번째 칸부터 $(N-1)$ 번째 칸 까지 데이터가 있는 배열에 아래의 과정을 수행한다
 1. 0~ $(N-1)$ 번째 칸에서 최소값을 구하여 0번째 칸과 위치를 변경한다.
 2. 1~ $(N-1)$ 번째 칸에서 최소값을 구하여 1번째 칸과 위치를 변경한다.
 3. ...
 4. $(N-2)$ ~ $(N-1)$ 번째 칸에서 최소값을 구하여 $(N-2)$ 번째 칸과 위치를 변경한다.

입력 형식

첫 줄에 전체 데이터의 수 N 이 주어진다. N 은 1,000이하의 자연수이다.

두번째 줄에 32비트 정수형 데이터가 공백으로 구분되어 총 N 개가 주어진다.

출력 형식

한 줄에 N 개의 데이터를 오름차순으로 정렬하여 출력한다. 각 데이터는 하나의 공백으로 구분되어 있어야 한다.

입/출력 예시

 : 공백  : 줄바꿈

예시 1

입력

5

3 5 1 2 4

출력

1 2 3 4 5

문제 11-선택정렬 (Selection Sort)

```
#include<stdio>
#include<iostream>

using namespace std;

/**
 * 주어진 범위의 최소값의 위치를 반환하는 함수
 * @param data : 데이터 배열
 * @param n : 배열의 크기
 * @param begin : 탐색 할 가장 첫 (왼쪽) 인덱스
 * @param end : 탐색 할 가장 마지막 (오른쪽) 인덱스
 * @return : data[begin] ~ data[end] 중 가장 작은 원소의 인덱스
 */
int getMinIndexInRange(int data[], int n, int begin, int end)
{

}

void selectionSort(int data[], int n)
{
    for (int i = 0; i < n; i++)
    {
        // 주어진 범위에서 가장 작은 원소의 위치를 찾는다.
        int minIndex = getMinIndexInRange(data, n, i, n - 1);

    }
}
```

```
int main()
{
    int n;
    int* data;

    scanf("%d", &n);
    data = new int[n];

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &data[i]);
    }

    selectionSort(data, n);

    for (int i = 0; i < n; i++)
    {
        if (i > 0)
        {
            printf(" ");
        }
        printf("%d", data[i]);
    }

    delete[] data;
    return 0;
}
```

문제 1J-합 구하기3

평소 머리가 나쁘다고 놀림을 자주 듣는 예인이는 몰래 수학 공부를 하고 있다. 수학 공부를 하던 예인이는 자연수에 대한 깊은 고찰을 하게 되었다. 어떤 자연수 N 이 있다고 했을 때 1이상 N 이하의 모든 자연수의 합은 손 쉽게 계산할 수 있다. 어떤 자연수 N 이 있다고 했을 때, 아래와 같은 식의 값은 어떻게 계산할 수 있을까?

$$(1\text{이상 } 1\text{이하의 모든 자연수의 합}) + (1\text{이상 } 2\text{이하의 모든 자연수의 합}) + \dots + (1\text{이상 } N\text{이하의 모든 자연수의 합})$$

어릴적부터 주술같은 주산을 배운 예인이는 위의 식을 계산하며 자신의 암산 능력을 다지기로 결정했다. 하지만 자신이 계산한 값이 정답이 맞는지 검증 할 수 없었던 예인이는 당신에게 자연수 N 을 입력하면 위의 식으로 계산한 결과를 알려주는 프로그램을 요구했다. 컴퓨터를 잘 다룬다고 주변에 소문난 당신은 귀여운 예인이를 위하여 프로그램을 작성해주기로 결정하였다.

입력 형식

첫 줄에 1이상 2,000이하의 자연수 N 이 공백없이 주어진다.

출력 형식

첫 줄에 입력 값 N 에 대하여 위의 식을 계산한 결과를 한 줄에 공백없이 출력한다.

입/출력 예시

 : 공백  : 줄바꿈

예시 1

입력

100



출력

171700



문제 1J-합 구하기3

```
#include<stdio>
#include<iostream>

using namespace std;

/**
 * 1부터 N까지의 자연수의 합을 계산하는 함수
 * @param i
 * @return
 */
int getRangeSumFromOne(int i) {

}

long long getAnswer(int N)
{

}

int main()
{
    int n;

    scanf("%d", &n);

    long long answer = getAnswer(n);

    printf("%lld\n", answer);

    return 0;
}
```

단원 정리

- 반복문의 실행 범위와 순서를 이해하고 설계하기
- 반복문을 활용하여 최대/최소값, 합, 특정 조건을 만족하는 숫자의 수와 위치를 탐색하기
- 반복문을 활용하여 배열의 숫자 정렬하기