

MID\_1

입력 받은 정수의 평균 구하기

5.21%

Description

중복된 숫자가 입력되기 전까지 양의 정수를 반복해서 입력 받고, 입력된 모든 정수의 평균을 출력하는 코드를 작성하시오.

-양의 정수만 입력되며 입력 값의 최대 개수는 100

-중복으로 입력된 값은 평균 계산에서 제외

-평균은 소수 둘째 자리까지 출력

Input

중복된 숫자가 입력되기 전까지 양의 정수를 반복해서 입력 받음

Output

입력된 모든 정수의 평균을 출력

단, 중복으로 입력된 값은 평균 계산에서 제외

```
#include <stdio.h>
int main() {
    int a[101] = {0, };
    int x, y = 0;
    int ans;

    while (1) {
        scanf("%d", &x);
        if (a[x] == 1)
            break;

        a[x] = 1;
        ans += x;
        y += 1;
    }

    printf("%.2f", (float)ans / y);
}
```

MID\_2

최대값 구하기

7.57%

#### Description

종료 조건까지 정수를 입력 받아, 최대값을 구하고, 최대값이 나타나는 위치(index)를 모두 출력하는 프로그램을 작성하시오.

-종료 조건: 0입력(최대값 계산 시 제외)

-최대 정수 개수: 100개

-같은 정수 입력 가능(중복 허용)

-정수는 한 개 이상 반드시 입력됨(즉, 종료 조건만 입력되는 경우는 고려하지 않음)

#### Input

- 0 이 입력될 때까지 정수를 입력 받음 (중복 허용)

-정수는 한 개 이상 반드시 입력됨(즉, 종료 조건만 입력되는 경우는 고려하지 않음)

#### Output

- 입력 받은 정수 중에서 최대값과 최대값이 나타나는 index를 출력

- 출력 형식: 첫 번째 줄에는 최대값을 출력하고, 두 번째 줄에는 각 최대값이 나타나는 위치(index)를 공백을 사이에 두고 차례대로 출력하되, 줄의 맨 처음에도 공백을 출력한다. (예: □3□6□7□11)

```
#include <stdio.h>
int main() {
    int x, max = -999999999, count = 0;
    int a[123] = {0, };

    while (1) {
        scanf("%d", &x);
        if (x == 0)
            break;

        if (max <= x && count) {
            max = x;
            a[count] = max;
        }
        count++;
    }

    printf("%d\n", max);
    for (int i = 0; i < 100; i++) {
        if (a[i] == max)
            printf(" %d", i);
    }
}
```

MID\_3

대칭 정수 만들기

3.93%

Description

종료 조건(0 입력)까지 양의 정수 N을 반복해서 입력 받는다. 입력 받은 정수 N 뒤에, N의 자리수를 역순으로 바꾼 정수를 붙여서, 대칭이 되는 정수 M을 만들어 출력하고, 대칭이 되는 정수 M의 각 자리수에서 5가 나오는 개수를 출력한다.

또한, 대칭되는 정수들 중 최소값과 최대값을 찾아 출력하는 프로그램을 작성하시오.  
배열 사용은 금한다!

Input

종료 조건(0 입력)까지 양의 정수를 반복해서 입력 받음

Output

대칭 정수와 대칭 정수에 포함된 5의 개수를 출력한 후,  
다음 줄에 최소값과 최대값을 출력

```
#include <stdio.h>
int main() {
    int x, y, temp;
    int max = -2147483648, min = 2147483647;

    while (1) {
        int count = 0;
        scanf("%d", &x);
        if (x == 0)
            break;

        temp = x;
        while (x > 0) {
            temp *= 10;
            temp += x % 10;
            x /= 10;
        }

        if (temp > max)
            max = temp;
        if (temp < min)
            min = temp;

        printf(" %d", temp);
        while (temp) {
            if (temp % 10 == 5)
                count++;
            temp /= 10;
        }
        printf(" %d", count);
    }
    printf("\n%d %d", min, max);
}
```

MID\_4

부분 배열의 합

1.83%

#### Description

입력 예시와 같이 N개의 정수를 입력 받는다 ( $N < 100$ ).

입력 받은 정수 중에서 음수가 아닌 정수가 연속적으로 나오는 부분 배열을 찾는다. 이러한 부분 배열 중에서 부분 배열의 합이 가장 큰 부분 배열을 출력하는 프로그램을 작성하시오.

단, 입력 받는 정수는 중복될 수 있으며, 부분 배열 중에서 합이 가장 큰 배열이 여러 개인 경우 배열의 길이가 가장 긴 배열을 출력한다. 만약, 이러한 배열 또한 여러 개라면, 가장 마지막에 나타나는 부분 배열을 출력한다.

#### Input

정수 N을 입력 받은 후, N개의 정수를 입력 받음

#### Output

입력 받은 정수 중에서 음수가 아닌 정수가 연속적으로 나오는 부분 배열 중에서 부분 배열의 합이 가장 큰 부분 배열을 출력

단, 부분 배열 중에서 합이 가장 큰 배열이 여러 개인 경우 배열의 길이가 가장 긴 배열을 출력  
만약, 이러한 배열 또한 여러 개라면, 가장 마지막에 나타나는 부분 배열을 출력

```

#include <stdio.h>
int main() {
    int a[100] = {0, };
    int n = 0, sum = 0, start = -1, end = -1;
    int ans_start = -1, ans_end = -1, max_sum = -2147483648, max_length = -1;

    scanf("%d", &n);

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

        if (a[i] >= 0 && (a[i-1] < 0 || i == 0)) {
            start = i;
            sum += a[i];
        }

        else if (a[i] >= 0 && a[i-1] >= 0) {
            sum += a[i];
        }

        else if (a[i] < 0 && i > 0) {
            end = i - 1;
            if (max_sum < sum && end - start >= max_length) {
                ans_start = start;
                ans_end = end;
                max_sum = sum;
                max_length = end - start;
                sum = 0;
            }
        }
    }

    if (sum > 0 && a[n-1] >= 0) {
        sum += a[n-1];
        if (max_sum <= sum && n-1 - start >= max_length) {
            ans_start = start;
            ans_end = n-1;
        }
    }

    if (ans_start >= 0 && ans_end >= 0)
        for (int i = ans_start; i <= ans_end; i++)
            printf("%d ", a[i]);
}

```

MID\_5

숫자 패턴 출력하기

11.22%

Description

양의 정수  $N$  ( $2 \leq N$ )을 입력 받아, 아래 예시와 같이  $N$ 개의 줄에 삼각형 모양으로 정수를 출력하는 프로그램을 작성하시오.

출력 시, %4d를 사용하시오.

Input

양의 정수  $N$  ( $2 \leq N$ )

Output

$N$ 개의 줄에 삼각형 모양으로 정수를 출력 (%4d를 사용)

```
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);

    for (int i = 0; i < n; i++) {
        for (int j = i*(i+1) / 2 + 1; j < (i+1)*(i+2) / 2; j++) {
            printf("%4d", j);
        }
        for (int j = (i+1)*(i+2) / 2; j > i*(i+1) / 2; j--) {
            printf("%4d", j);
        }
        printf("\n");
    }
}
```



MID\_6

소수 출력하기

5.95%

#### Description

종료 조건 시까지 양의 정수 N을 반복해서 입력 받아, 입력 받은 정수 중 소수를 출력하고, 출력된 소수의 다음 소수를 함께 출력하는 프로그램을 작성하시오. 소수는 1과 자신 이외의 수로 나누어 떨어지지 않는 수이다. 입력된 첫 번째 정수가 음수이거나, 입력 받은 정수 중 소수가 존재하지 않는 경우에는 “none”을 출력한다.

-종료 조건: 음수 입력

- 0과 1은 입력되지 않는다고 가정한다.

#### Input

음수가 입력될 때까지 양의 정수 N이 입력

#### Output

-소수를 출력하고, 출력된 소수의 다음 소수를 함께 출력

- 입력된 첫 번째 정수가 음수이거나, 입력 받은 정수 중 소수가 존재하지 않는 경우에는 “none”을 출력

```
#include <stdio.h>

int is_prime(int n) {
    for (int i = 2; i < n-1; i++)
        if (n % i == 0)
            return 0;
    return 1;
}

int next_prime(int n) {
    while (1)
        if (is_prime(++n))
            return n;
}

int main() {
    int n, count = 0;
    while (1) {
        scanf("%d", &n);
        if (n < 0)
            break;

        if (is_prime(n)) {
            printf("%d %d\n", n, next_prime(n));
            count++;
        }
    }

    if (count == 0)
        printf("none");
}
```