

■ 문제(Insertion 정렬 수정)

수업시간에 배운 insertionSort의 코드는 다음과 같다. 즉 배열과 그 크기를 매개변수로 받아서 배열을 정렬한다. 아래의 코드를 참고하여 문제를 풀라.

```
insertionSort(int* A, int n) //배열 A을 정렬한다
{
    int i, j, k, temp;

    for (i = 1 ; i < n ; i++) {
        for (j = 0; j < i; j++)
            if (A[j] > A[i]) break;

        temp = A[i];
        for (k = i; k > j; k--)
            A[k] = A[k-1];
        A[j] = temp;
    }
}
```

이를 배열과 배열의 첫인덱스, 마지막인덱스를 매개변수로 받도록 수정한 insertionSort2 함수를 정의하여 정렬하는 프로그램을 완성하라. 정렬할 배열의 크기는 100 이하라고 가정한다.

실행예:

입력)

```
10 <- 배열의 크기
9 3 5 7 2 4 6 8 1 0 <- 배열의 원소 값
0 9 <- 정렬할 시작 인덱스와 끝 인덱스
```

출력)

```
0 1 2 3 4 5 6 7 8 9
```

실행예:

입력)

```
10 <- 배열의 크기
9 3 5 7 2 4 6 8 1 0 <- 배열의 원소 값
1 8 <- 정렬할 시작 인덱스와 끝 인덱스
```

출력)

```
9 1 2 3 4 5 6 7 8 0
```

<pre>void insertionSort2(int *A, int left, int right) { <i>int i, j, k, temp;</i> <i>for (i = left + 1; i <= right ; i++)</i> { <i>for (j = left; j < i; j++)</i> <i>if (A[j] > A[i]) break;</i> <i>temp = A[i];</i> <i>for (k = i; k > j; k--)</i> <i>A[k] = A[k-1];</i> <i>A[j] = temp;</i> } } #include <stdio.h> void printList(int A[], int n) { int i; for (i = 0; i < n; i++) printf("%d ", A[i]); printf("\n"); }</pre>	<pre>int main(void) { int list[10]; scanf("%d", &n); for (insertionSort2(list, 0, 9); printList(list, 10);) }</pre>
--	--

■ 문제(암호화)

다음과 같이 암호화표를 완성하는 프로그램을 만들려한다.
distance 값을 입력받아
distance 번째에 A를 넣고
다시 거기서 distance 떨어진 곳에 다음 알파벳인 B를 넣고 이를 반복한다.

즉, distance가 5라면, 다섯번째 문자부터 distance의 간격으로 A, B, C를 넣는다. 암호화 표의 중간 모습은 다음과 같다.

알파벳	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
암호 알파벳					A					B					C					D					E	

그리고는 표의 앞에서부터 나머지 알파벳들을 차례로 넣어서 아래와 같은 표를 만든다.

알파벳	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
암호 알파벳	F	G	H	I	A	J	K	L	M	B	N	O	P	Q	C	R	S	T	U	D	V	W	X	Y	E	Z

이렇게 생성된 암호 알파벳 시퀀스를 출력하는 프로그램을 작성하라.

실행예:
입력)
5
출력)
FGHIAJKLMBNOPQCRSTUDVWXYEZ

■ 문제(재귀 & Memoization)

아래의 재귀적 정의를 보고 입력된 정수 $n(>0)$ 에 대해서 H_n 을 *빠르게* 구하는 프로그램을 완성하시오.

$$H_n = 1 \text{ (if } n = 1)$$

$$= (n - 1) \times H_{n-1} + (n - 2) \times H_{n-2} + \dots + 2H_2 + 1 \times H_1 \text{ (if } n > 1)$$

실행예:

입력) 1

출력) 1

실행예:

입력) 4

출력) 12

```
#include <stdio.h>
#include <stdlib.h>
int H(int n, int *M)
{
    .....
    .....
    .....
}

int main(void)
{
    int n;
    int *M;
    int i;
    scanf("%d", &n);

    // 메모 M에 대한 동적 메모리 할당
    .....
    .....
    .....
    .....

    printf("%d\n", H(n, M));
    return 0;
}
```

■ 문제(문해기 과제 난이도)

(문해기 과제 난이도) 문제해결기법 교수는 과목 숙제에 대한 학생들의 체감 난이도를 알고 싶어졌다. 그래서 운영하고 있는 숙제 관리 시스템을 조금 수정하기로 했고 N개의 숙제에 대해 M명의 학생들이 얼마나 숙제를 하였는지를 출력하는 프로그램을 추가 구현하였다. 이 프로그램은 한 줄에 학생 1명의 M개의 숙제에 대한 결과를 출력하며 각 숙제에 대해 풀었으면 1로 못 풀었으면 0으로 표시한다.

아래는 이 프로그램으로 4개의 숙제를 5명의 학생들이 푼 결과 화면이다.

4 5 → 4개의 숙제에 대한 5명의 결과

1 0 0 0 → 첫 번째 학생은 1번 숙제를 풀었으며 2/3/4번 숙제는 풀지 못 했다.

1 0 1 0 → 두 번째 학생은 1/3번 숙제를 풀었으며 2/4번 숙제는 풀지 못 했다.

1 0 1 0 → 세 번째 학생은 1/3번 숙제를 풀었으며 2/4번 숙제는 풀지 못 했다.

1 0 0 1 → 네 번째 학생은 1/4번 숙제를 풀었으며 2/3번 숙제는 풀지 못 했다.

1 1 1 1 → 다섯 번째 학생은 모든 숙제를 풀었다.

위의 화면을 입력으로 하면 1번 문제는 못 푼 학생이 없고, 2번 문제는 1명이, 3번 문제는 3명이 4번 문제는 2명이 못 풀었다. 따라서 2번 문제가 가장 난이도가 높으며 1번 문제가 난이도가 제일 낮다.

□ 난이도-1

문제당 숙제를 제출한 학생수를 오름차순으로 출력하는 프로그램을 작성하라.

입력)

4 5

1 0 0 0

1 0 1 0

1 0 1 0

1 0 0 1

1 1 1 1

출력)

1 2 3 5 <- 문제2, 문제4, 문제3, 문제1에 대해서 각각 1,2,3,5명의 학생이 제출하였다

□ 난이도-2

위와 같은 입력을 받아 난이도가 어려운 순서대로 문제 번호를 출력하는 프로그램을 작성하시오. 문제의 난이도는 못 풀어낸 학생의 수가 많을수록 높다.

따라서 출력은 2 4 3 1로 되어야 한다. 못 푼 학생의 수가 같은 문제가 여러 개가 있으면 문제 번호가 작은 것을 먼저 출력한다.

<p>실행예:</p> <p>입력)</p> <p>4 5</p> <p>1 0 0 0</p> <p>1 0 1 0</p> <p>1 0 1 0</p> <p>1 0 0 1</p> <p>1 1 1 1</p> <p>출력)</p> <p>2 4 3 1 <- 문제 2가 가장 어렵고 4, 3, 1의 순으로 난이도를 계산했다</p>	<p>실행예:</p> <p>입력)</p> <p>5 6</p> <p>1 1 0 0 0</p> <p>1 0 1 0 0</p> <p>1 1 1 1 0</p> <p>1 1 0 1 0</p> <p>1 0 0 1 0</p> <p>1 0 1 1 1</p> <p>출력)</p> <p>5 2 3 4 1 <- 문제 2와 3은 똑같이 3명의 숙제를 제출하여 난이도가 같으나 번호가 작은 문제를 먼저 출력한다</p>
--	---

■ 문제(피구)

N명(N은 짝수)의 컴퓨터학과 학생들이 MT를 가서 저녁 먹기 전까지 시간이 조금 남아서 2팀으로 나누어서 피구를 하기로 했다. MT를 함께한 지도 교수는 게임을 재미있게 하기 위해서 다음과 같이 팀을 나누기로 했다. 지도 교수는 미리 조사된 학생 개개인의 능력치를 이용하여 두 팀 사이의 능력치의 합을 최소로 하도록 팀을 구성하기로 하는 것이 최종 목적이다.

예를 들어 6명의 학생에 대해 아래와 같이 능력치를 입력 받는다면

1 2 3 4 3 2

(1~6번 학생의 능력 각각 1, 2, 3, 4, 3, 2이다.)

□ 피구-1

N명의 학생에서 팀을 결성하였을때 가능한 팀 결성과 각 팀의 능력치의 합을 출력하는 프로그램을 작성하라. 위의 예에 대한 출력 결과는 다음과 같다.

```
1 2 3 : 60
1 2 4 : 70
1 2 5 : 60
1 2 6 : 50
1 3 4 : 80
1 3 5 : 70
1 3 6 : 60
1 4 5 : 80
1 4 6 : 70
1 5 6 : 60
2 3 4 : 90
2 3 5 : 80
2 3 6 : 70
2 4 5 : 90
2 4 6 : 80
2 5 6 : 70
3 4 5 : 100
3 4 6 : 90
3 5 6 : 80
4 5 6 : 90
```

□ 피구-2

1/3/4번 학생이 하나의 팀을 이루고 2/5/6번 학생이 다른 팀을 이룬다면 각 팀의 총 능력치는 $80(=10+30+40)$ 그리고 $70(=20+30+20)$ 이 된다. 그리고 이 경우일 때가 팀 능력치의 차이가 최소가 되어 가장 재미있는 피구를 하게 될 것이다.

두 팀의 능력치 차이의 최소값을 구하는 프로그램을 작성하시오.

입력)

6 (→ 6명의 학생)

10 20 30 40 30 20 (→ 학생들의 개별 능력치)

출력)

10