

프로그래밍 과제 08

1. 다음의 순환 함수의 반환값을 x 와 y 의 함수로 나타내면?

```
int fun1(int x, int y)
{
    if (x > y)
        return 0;
    return y + fun1(x, y-1);
}
```

x 부터 y 까지의 합

2. 다음의 순환함수의 반환값을 n 의 함수로 나타내면?

```
/* Assume that  $n \geq 1$  */
int fun2(int n)
{
    if(n == 1)
        return 0;
    else
        return 1 + fun2(n/2);
}
```

$\lceil \log_2 n \rceil$, $\lceil \cdot \rceil$ 는 가우스 기호

3. 다음의 순환함수가 결과적으로 하는 일은?

```
void fun3(int n)
{
    if (n == 0)
        return;
    fun3(n/2);
    printf("%d", n%2);
}
```

2진수 변환출력

4. 다음의 순환함수가 결과적으로 하는 일은?

```
void fun4(int n)
{
    if (n > 1)
        fun4(n-1);
    for (int i = 0; i < n; i++)
        printf(" * ");
}
```

1부터 n 까지의 합만큼의 asterisk 출력

5. 다음의 함수 fun5의 반환값을 a 와 b 에 관한 식으로 표현하면?

```
int fun(int x, int y)
{
    if (y == 0) return 0;
    return (x + fun(x, y-1));
}

int fun5(int a, int b)
{
    if (b == 0) return 1;
    return fun(a, fun5(a, b-1));
}
```

fun : return $x * y$
fun5 : a^b

6. 다음 함수가 결과적으로 하는 일을 최대한 간명하게 설명하라.

```
int fun6(int a[], int n)
{
    if(n == 1)
        return a[0];

    int x = fun6(a, n-1);
    return (x > a[n-1] ? x : a[n-1]);
}
```

return max value of list a

7. 다음 함수가 결과적으로 하는 일을 최대한 간명하게 설명하라.

```
double fun7(double a[], int n)
{
    if (n==1) return a[0];
    else
        return (a[n-1] + (n-1)*fun7(a, n-1))/n;
}
```

return average of list a

8. 다음 함수가 결과적으로 하는 일을 최대한 간명하게 설명하라.

```
int fun8(int a, int b)
{
    if (b == 0)
        return 1;
    if (b % 2 == 0)
        return fun8(a*a, b/2);
    return fun8(a*a, b/2)*a;
}
```

```
return a^b
```

9. 다음 함수가 결과적으로 하는 일을 최대한 간명하게 설명하라.

```
void fun9(int arr[], int start_index, int end_index)
{
    if(start_index >= end_index)
        return;
    int min_index;
    int temp;

    /* Assume that minIndex() returns index of minimum value in
       array arr[start_index...end_index] */
    min_index = minIndex(arr, start_index, end_index);

    temp = arr[start_index];
    arr[start_index] = arr[min_index];
    arr[min_index] = temp;

    fun9(arr, start_index + 1, end_index);
}
```

partial sort array arr start_index to end_index

10. [프로그래밍 과제] 길이가 같은 두 이진수열간의 “거리”는 두 이진수열에서 서로 다른 비트의 개수로 정의된다. 예를 들어 001100과 101010간의 거리는 3이고, 0001과 0010의 거리는 2이다. 입력으로 길이가 N (≤ 16)인 하나의 이진수열과 하나의 정수 K ($\leq N$)를 받은 후 입력된 이진수열과의 거리가 K 인 모든 이진수열을 찾아 출력하는 프로그램을 작성하라. 예를 들어 $N = 4$, $K = 2$ 이고 입력된 이진수열이 0000이라면 다음의 이진수열들을 출력해야 한다.

0011 0101 0110 1001 1010 1100

입력 형식

입력은 키보드로부터 받는다. 입력의 첫 줄에는 두 정수 N 과 K 가 주어진다. 다음 줄에는 길이가 N 인 이진수열이 주어진다.

출력 형식

화면으로 답을 출력한다.

11. [프로그래밍 과제] 처음에는 모든 양의 정수들의 집합에서 시작한다.

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19,

우선 모든 두 번째 수를 제거한다. 그러면 다음의 수들이 남는다.

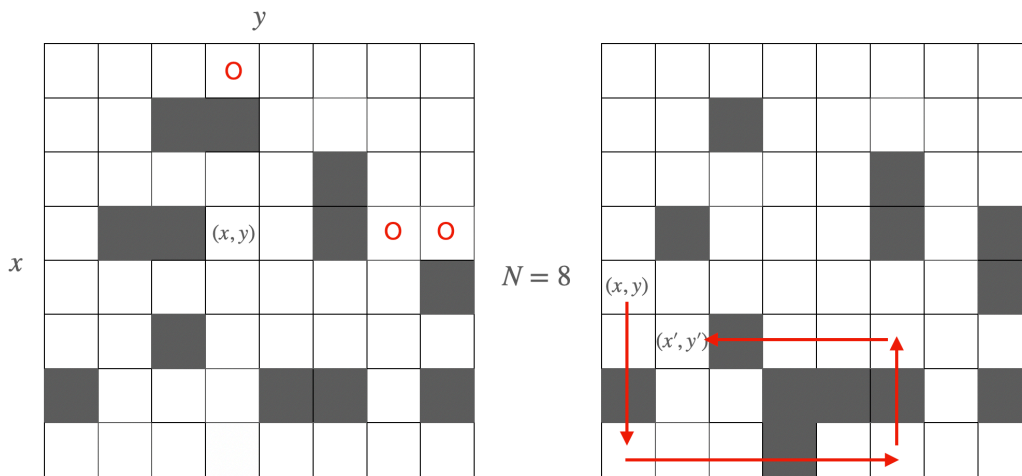
1, 3, 5, 7, 9, 11, 13, 15, 17, 19,

이번에는 여기에서 모든 세번째 숫자들을 제거한다.

1, 3, 7, 9, 13, 15, 19,.....

그 다음에는 모든 네번째 수를 제거한다. 이 과정을 무한히 계속한다. 결국 제거되지 않고 남아있는 수들을 “행운수”라고 부른다. 예를 들면 1, 3, 7, 13,... 등이 행운수이다. 입력으로 하나의 정수 n 을 받아서 행운수인지 아닌지 검사하여 yes 혹은 no를 출력하는 프로그램을 작성하라. n 은 1,000,000이하이다.

12. [프로그래밍 과제] $N \times N$ 크기의 장기판이 있다. 체스처럼 말들이 셀(cell)에 놓인다고 가정하자. 장기의 말들 중에 포(包)는 다음과 같은 규칙으로 움직인다: 위치 (x, y) 에 놓인 포(包)는 상하좌우 네 방향 중 하나로 다른 말을 하나 건너뛰어 어떤 위치로든 한 번에 이동할 수 있다. 이때 이동 거리는 상관없다. 아래의 왼쪽 그림에서 (x, y) 에 놓인 포(包)는 빨간 원(O)으로 표시된 세 지점 중 하나로 한 번에 이동할 수 있다. 장기판의 현재 상태와 현재 포(包)가 놓여있는 위치 (x, y) 와 최종적으로 이동할 목표위치 (x', y') 를 입력 받은 후, 현재위치에서 목표 위치로 이동하는 방법이 존재하는지 검사하여 Yes 혹은 No를 출력하는 프로그램을 작성하라. 아래의 오른쪽 그림은 출발점이 $(4, 0)$ 이고 목표점이 $(5, 1)$ 인 경우이다. 이 경우 빨간 화살표로 표시된 것 처럼 4번 움직여서 목표점에 도착할 수 있다.



입출력 형식:

입력은 input12.txt 파일로 주어진다. 파일의 첫 줄에는 장기판의 크기 $N \leq 16$ 이 주어지고, 이어진 N 줄에는 장기판의 상태가 주어진다. 빈 칸은 0, 이미 다른 말이 놓여 있는 칸은 1로 표시된다. 그런 다음 출발점의 좌표와 도착점의 좌표가 각각 주어진다. 출력은 화면으로 Yes 혹은 No를 출력한다.

입력 파일의 예	출력 예
<pre> 8 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0 1 0 0 0 0 2 2 4 3 </pre>	No

8 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0 1 0 0 0 0 3 3 6 2	Yes
8 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 3 3 1 5	No
8 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 3 3 6 2	Yes
8 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 3 3 4 6	No