제2장 연습문제

[연습문제]

1. 다음의 순환 함수의 반환값을 x와 y의 함수로 나타내면?

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

2. 다음의 순환함수가 결과적으로 하는 일은?

```
void fun(int n)
{
  if (n == 0)
    return;

  printf("%d", n%2);
  fun(n/2);
}
```

3. 다음의 함수의 반환값을 x와 y에 관한 식으로 표현하면?

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

4. 다음의 함수 fun2의 반환값을 a와 b에 관한 식으로 표현하면?

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

int fun2(int a, int b)
{
    if (b == 0) return 1;
    return fun(a, fun2(a, b-1));
}
```

5. 다음 함수가 하는 일을 설명하라.

```
int fun(unsigned int n)
{
    if (n == 0 || n == 1)
        return n;
    if (n%3 != 0)
        return 0;
    return fun(n/3);
}
```

6. 다음 프로그램의 실행 결과를 예측하라.

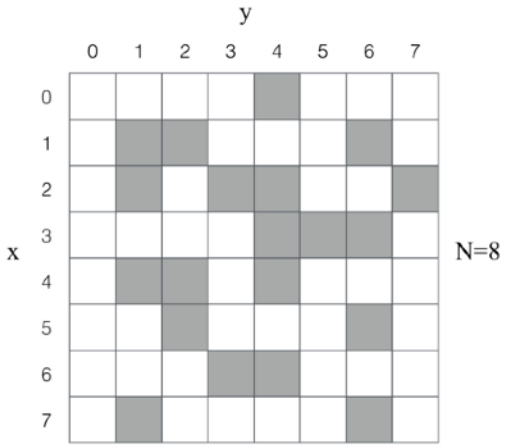
```
#include <stdio.h>
int f(int n)
{
    if(n <= 1)
        return 1;
    if(n%2 == 0)
        return f(n/2);
    return f(n/2) + f(n/2+1);
}


int main()
{
    printf("%d", f(11));
    return 0;
}
```

7. 강의 슬라이드의 미로찾기 알고리즘이 올바르다는 것을 수학적 귀납법으로 증명하라. 올바르다는 것을 보이기 위해서는 경로가 존재하면 true를 반환하고, 존재하지 않을 경우 false를 반환하고 종료한다는 것을 보여야 한다.

[프로그래밍 과제]

1. 미로 찾기 문제에서 입구에서 출구까지 가는 서로 다른 경로의 개수를 계산하여 출력하는 프로그램을 순환함수(recursion)를 이용하여 작성하라. 단 같은 위치를 2번 이상 방문하는 경로는 카운트하지 않는다. 입구의 위치는 (0,0), 출구는 (N-1,N-1)이다. N은 8 이하이다. (경로의 개수가 int의 표현범위를 넘는 경우는 없다고 가정해도 된다.)



8
0 0 0 0 1 0 0 0
0 1 1 0 0 0 1 0
0 1 0 1 1 0 0 1
0 0 0 0 1 1 1 0
0 1 1 0 1 0 0 0
0 0 1 0 0 0 1 0
0 0 0 1 1 0 0 0
0 1 0 0 0 0 1 0

입력 형식

입력 파일의 이름은 input1.txt이다. 입력 파일의 첫 줄에는 테스트 케이스의 개수 T≤10가 주어지고, 이어서 T개의 테스트 데이터가 주어진다. 각 테스트 케이스의 첫 줄에는 미로의 크기 N이 주어진다. 이어진 N 줄에는 각 줄마다 N개의 0 혹은 1이 한 칸씩 띄어져서 주어진다. 0은 통로, 1은 지나갈 수 없는 벽을 표시한다.

출력 형식

각 테스트 케이스마다 경로의 개수를 화면으로 출력한다.

Sample Input	**Output for the Sample Input**
5 // T=5 8 // N=8 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 1 0 1 1 0 0 1 0 0 0 0 1 1 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 3 // N=3 0 0 0 0 0 0 0 0 0 3 // N=3 0 0 0 0 1 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0	6 12 2 184 128

2. 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 이진수열 s와 하나의 정수 K(≤N)를 받은 후 s와의 거리가 K인 모든 이진수열을 찾아 출력하는 프로그램을 작성하라. 예를 들어 N=4, K=2, s=0000이라면 다음의 이진수열들을 출력해야 한다.

0011 0101 0110 1001 1010 1100

입력 형식

입력 파일의 이름은 input2.txt이다. 입력 파일의 첫 줄에는 테스트 케이스의 개수 T≤10가 주어지고, 이어서 T개의 테스트 데이터가 주어진다. 각 테스트 케이스의 첫 줄에는 두 정수 N과 K가 주어진다. 각 테스트 케이스의 두번째 줄에는 길이가 N인 이진수열이 주어진다.

출력형식

output2.txt라는 이름의 파일로 각 테스트 케이스에 대한 답을 출력한다.

Sample Input	**Output for the Sample Input (output2.txt)**
2 4 2 1000 6 6 000000	0100 0010 0001 1110 1011 1101 111111

3. 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 n을 받아서 행운수인지 아닌지 검사하여 yes 혹은 no를 출력하는 프로그램을 작성하라. n은 1,000,000이하이다.

입력 형식

입력 파일의 이름은 input3.txt이다. 입력 파일의 첫 줄에는 테스트 케이스의 개수 T≤10가 주어지고, 이어서 T개의 양의 정수가 한 줄에 하나씩 주어진다.

출력형식

화면으로 각 테스트 케이스에 대한 답을 출력한다.

Sample Input	**Output for the Sample Input**
7 13 19 221 223 707 976243 999666	yes yes no yes no yes no