

프로그래밍 과제 01

1. 입력으로 하나의 정수 $2 \leq n \leq 30$ 을 받아서 Fibonacci 수 f_n 을 계산하여 출력하는 프로그램을 작성하라. Fibonacci 수열은 다음과 같이 정의된다.

$$f_0 = 1$$

$$f_1 = 1$$

$$f_n = f_{n-1} + f_{n-2}, \quad n > 1$$

입력 예	출력
5	8
10	89
20	10946
30	1346269

2. 입력으로 하나의 양의 정수 n 을 받은 후 다음의 합을 구하여 출력하는 프로그램을 작성하라. 소수점 4째 자리 이후에는 아래의 예시와 달라도 무시하라.

$$1 - \frac{1}{2} + \frac{1}{2^2} - \frac{1}{2^3} + \cdots + (-1)^n \frac{1}{2^n}$$

입력 예	출력
1	0.5000
2	0.7500
5	0.6562
10	0.6670
20	0.6667

3. 입력으로 하나의 양의 정수 n 을 받은 후 다음의 합을 구하여 출력하는 프로그램을 작성하라. 소수점 4째 자리 이후에는 아래의 예시와 달라도 무시하라.

$$1 + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!}$$

입력 예	출력
1	1.0000
2	1.5000
5	1.7167
10	1.7183
12	1.7183

4. 입력으로 세 정수 a, b, c 를 받은 후 오름차순으로 정렬하여 출력하는 프로그램을 작성하라. 프로그램은 다음의 형태를 유지해야 한다. (즉, 단지 오름차순으로 출력만 하는 것이 아니라 실제로 세 변수의 값이 크기 순으로 정렬되어야 한다.)

```
#include <iostream>
using namespace std;
```

```

int main()
{
    int a, b, c;
    cin >> a >> b >> c;

    /* 이 부분에서 변수 a, b, c의 값을 서로 교환하여 a<=b<=c가 되도록 만들어라. */

    cout << a << b << c << endl;
    return 0;
}

```

5. 먼저 양의 정수 N 을 입력받고 이어서 N 개의 정수를 키보드로 부터 입력받는다. 입력된 정수들 중에서 음이 아닌 정수들 중에서 최소값을 찾아서 출력하는 프로그램을 작성하라. 그런 값이 존재하지 않을 경우 -1 을 출력하라.

입력 예	출력
10 1 4 8 11 -4 2 9 13 3 20	1
12 -2 -5 -6 -3 -10 -11 -7 -8 -9 -10 -9 -18	-1
20 -7 -8 -9 -12 0 -7 19 2 19 20 7 -5 -61 7 18 27 -81 77 16 9	0
8 -10 -7 1 7 2 4 8 8	1

6. 먼저 양의 정수 N 을 입력받고 이어서 N 개의 정수를 키보드로 부터 입력받는다. 새로운 정수가 입력될 때 마다 현재까지 입력된 정수들 중에서 최대값과 최소값의 차이를 계산하여 출력하는 프로그램을 작성하라. 배열을 사용해서는 안된다.

입력 예	출력
10 1 4 8 11 -4 2 9 13 3 20	0 3 7 10 15 15 15 17 17 24
12 -10 7 0 7 2 4 8 8 8 1 10 -15	0 17 17 17 17 17 18 18 18 18 20 25

7. 두 개의 날짜를 입력받은 후 그 중 첫 번째 날짜가 더 이르면 -1 , 두 번째 날짜가 더 이르면 1 , 두 날짜가 동일하면 0 을 출력하는 프로그램을 작성하라. 각각의 날짜는 년, 월, 일을 나타내는 3개의 정수로 주어진다. 날짜의 유효성을 체크할 필요는 없다.

입력 예	출력
2017 1 3 2016 12 9	1
2017 3 3 2017 3 3	0
2017 2 5 2017 2 4	1
2017 3 9 2017 2 19	1
1999 3 3 2011 5 8	-1

8. 두 개의 폐구간 $[a, b]$ 와 $[c, d]$ 를 입력받는다. a, b, c, d 는 모두 정수이고, $a \leq b$ 이고 $c \leq d$ 이다. 두 구간이 겹치는지 판단하여 Yes 혹은 No로 출력하는 프로그램을 작성하라. 두 구간이 겹친다는 것은 교집합이 공집합이 아니라는 의미이다.

입력 예	출력
1 3 4 6	No
4 7 1 2	No

입력 예	출력
4 8 1 4	Yes
1 8 3 5	Yes
4 7 2 12	Yes
1 8 8 12	Yes

9. 입력으로 변수 x 와 y 에 관한 다항식의 한 항의 계수, 변수 x 의 차수, 그리고 변수 y 의 차수를 받아서 그 항을 출력하는 프로그램을 작성하라. 계수는 정수이고 x 와 y 의 차수는 음이 아닌 정수이다. 출력은 아래의 입출력 예와 같은 규칙을 따라야 한다 (가령 계수 1은 출력하지 않고, -1은 -부호만 출력한다. 등)

```
int main()
{
    int coef, expo_x, expo_y;
    cin >> coef >> expo_x >> expo_y;
    /* 이 부분에 출력하는 코드를 추가하라 */
    return 0;
}
```

입력 예	출력 예
1234 23 12	1234x ²³ y ¹²
-1112 21 1	-1112x ²¹ y
1 211 31	x ²¹¹ y ³¹
-1 1 12	-xy ¹²
-1 31 0	-x ³¹
1 0 12	y ¹²
-135 1 1	-135xy
91 0 0	91
1 0 0	1
-1 0 0	-1

10. 입력으로 하나의 양의 정수 N 을 받은 후 이진수로 변환하여 출력하는 프로그램을 작성하라. 프로그램의 골격은 다음과 같다.

```
#include <iostream>
using namespace std;
int main() {
    int N;
    cin >> N;

    /* 여기에서 N보다 작거나 같으면서 가장 큰 2의 거듭제곱수 v를 구한다. */
    while (v > 0) {
        /* N ≥ v이면 N = N-v가되고 */
        /* 1을 출력한다. 그렇지 않으면 */
        /* 0을 출력한다. 두 경우 모두 */
        /* v는 1/2한다. */
    }
}
```

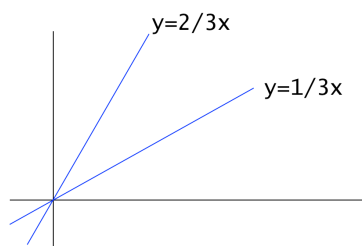
입력 예	출력
123	1111011

입력 예	출력
1111111	100001111010001000111
1024	10000000000
1	1

11. 사용자로 부터 하나의 양의 정수 n 을 입력 받은 후 자리를 뒤집어서 출력하는 프로그램을 작성하라. 예를 들어 $n = 12345$ 이면 54321을 출력하면 된다.
12. 키보드로 부터 2개 이상의 정수들을 연속해서 입력받는다. -1은 입력의 끝을 의미한다 (즉, -1 자체는 입력된 정수로 간주하지 않는다.) 입력된 정수들 중에서 극소값(local minima)의 개수를 카운트하여 출력하라. 여기서 극소값이란 바로 직전에 입력된 정수보다 작거나 같고, 바로 다음에 입력된 정수보다 작거나 같은 수를 의미한다. 단, 첫 번째 정수는 2번째 정수보다 작거나 같으면 극소값으로 간주하고, 마지막 정수는 바로 이전의 정수보다 작거나 같으면 극소값으로 간주한다.

입력 예	출력
1 1 2 1 3 5 9 8 -1	4
1 2 -1	1
9 7 5 7 8 1 3 9 -1	2
1 2 5 4 5 2 7 8 1 1 1 1 2 -1	7

13. 2차원 평면의 1사분면에서 직선 $y = 2/3x$ 의 아래쪽에 있고 직선 $y = 1/3x$ 의 위쪽에 있으면서 원점으로부터 거리가 100이하인 정수 좌표 점의 개수는? x -좌표나 y -좌표가 0이거나 직선 상에 있는 점도 포함한다. (정답: 1362)



14. 2에서 100 사이의 정수들 중에서 서로 서로소의 관계인 세 정수쌍의 개수를 세는 프로그램을 작성하라. 예를 들어 (3, 5, 7)은 서로 서로소이다. (정답: 44142)
15. 1에서 N 사이의 정수들을 모두 종이에 적었을 때 사용된 0의 개수가 총 몇개인지 계산하는 프로그램을 작성하라.

입력 예 (N)	출력
10	1
100	11
1234	343
10120	3143
100000	38894

16. 1693년에 Samuel Pepys는 뉴턴에게 “주사위를 6번 던져서 적어도 한 번 1이 나오는 것”과 “주사위를 12번 던져서 적어도 두 번 1이 나오는 것” 중 어느 쪽이 일어날 확률이 더 높은지 질문하였다. 이 질문에 대한 답을 실험을 통해 알아내는 프로그램을 작성하라. 즉 $T=1,000,000$ 번의 실험을 하여 두 사건이

일어난 경우의 수를 각각 카운트하는 프로그램을 작성하라. (6번 던져서 적어도 1번 1이 나오는 사건의 확률은 $1-(5/6)^6 \approx 0.6651$ 이고, 12번 중 적어도 2번 1이 나오는 사건의 확률은 $1-(5/6)^{12}-2(5/6)^{11} \approx 0.618667$ 이다. 대략 이 확률에 근접하는 실험 결과가 나오면 정답이다.)

17. 먼저 입력될 정수의 개수 $n \leq 100$ 을 입력받고, 이어서 n 개의 정수를 받아 평균과 표준편차를 계산하여 출력하는 프로그램을 작성하라. 표준편차는 다음과 같이 정의된다. 루트(square root)를 계산하기 위해서 `<cmath>`를 `include`하고 `sqrt`함수를 사용하라.

$$SD = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

입력 예	출력
3 1 1 1	1.0 0.0
6 1 9 8 2 5 12	6.1667 3.8909
6 -5 -1 -4 -1 -12 -9	-5.3333 4.0277
8 1 2 3 4 5 6 7 8	4.5 2.2913
10 1 4 0 4 4 -11 4 1 1 6	1.4 4.521

18. 하나의 양의 정수 N 을 입력 받은 후 정수 N 에 0에서 9까지의 digit가 각각 몇 번씩 등장하는지 카운트하여 순서대로 출력하는 프로그램을 작성하라. 예를 들어 $N = 120341279$ 라면 1, 2, 2, 1, 1, 0, 0, 1, 0, 1이라고 출력한다. 즉, 0은 1번, 1은 2번, 2는 2번, 3과 4는 1번, 그리고 7과 9가 1번씩 등장했고, 5, 6, 8은 한 번도 등장 하지 않았다는 의미이다.
19. 먼저 입력될 정수의 개수 $n \leq 100$ 을 입력받고, 이어서 n 개의 정수를 받아 순서대로 배열에 저장한다. 그런 다음 키보드로 부터 다시 하나의 정수 k 를 입력받은 후 n 개의 정수들 중에 서 k 에 가장 가까운, 즉 k 와의 차이의 절대값이 가장 작은 정수를 찾아 출력하는 프로그램을 작성하라.

입력 예	출력
6 1 9 8 2 5 12	8
7 6 -5 -1 -4 -1 -12 -9	-1
20 8 1 2 3 4 5 6 7 8	5
5 10 1 4 0 4 4 -11 4 1 1 6	-11
-8 12 73 28 1 9 37 46 -92 -8 37 0 0 12	28
24	

20. 먼저 입력될 정수의 개수 $n \leq 100$ 을 입력받고, 이어서 n 개의 정수를 받아 순서대로 배열에 저장한다. 이어서 추가로 하나의 양의 정수 $k < n$ 를 입력받는다. 배열에 저장된 모든 값들을 k 칸씩 뒤로 이동한다. 이때 배열의 맨 끝과 시작은 원형으로 맞닿아 있다고 가정한다. (즉, 맨 마지막 값을 한 칸 이동하면 배열의 맨 앞으로 온다. 이것을 circular right shift라고 부른다.) 예를 들어 $n = 4$ 이고 정수들이 1,2,3,4

이고 $k = 2$ 라면 결과는 3,4,1,2가 된다. 이렇게 이동 한 후 모든 정수들을 배열에 저장된 순서로 출력하라.

입력 예 (INPUT7.TXT)	출력
5 1 2 3 4 5 3 10 1 2 3 4 5 6 7 8 9 10 6	3 4 5 1 2 5 6 7 8 9 10 1 2 3 4

21. 먼저 입력될 정수의 개수 $n \leq 100$ 이 주어지고, 이어서 n 개의 정수가 주어진다. 정수들 중에서 자신보다 먼저 나온 모든 정수들 보다 크거나 같은 정수를 리더(**leader**)라고 부른다. 그리고 첫 번째 정수는 무조건 리더이다. 리더가 아닌 모든 정수들은 무시하고 리더들만 입력된 순서대로 배열에 저장하는 프로그램을 작성하라. 리더들은 배열의 맨 앞에서 부터 빈 칸 없이 저장되어야 한다. 먼저 리더들의 개수를 출력하고 콜론(:)을 출력한 후 이어서 리더들을 순서대로 화면으로 출력하라.

입력 예 (INPUT9.TXT)	출력
10 1 2 3 4 5 6 7 8 9 10 5 1 1 1 1 1 12 10 1 4 0 4 4 -11 4 1 1 6 -8 9 6 -5 -1 -4 20 -1 -12 -9 20 14 12 0 28 1 9 37 46 -92 -8 37 0 0 12 124	10: 1 2 3 4 5 6 7 8 9 10 5: 1 1 1 1 1 1: 10 3: 6 20 20 5: 12 28 37 46 124

22. 매일 매일의 주식 가격을 표현하는 $N \leq 100$ 개의 양의 정수가 주어진다. -1이 입력되면 입력이 끝났음을 의미한다. 주식 거래로 얻을 수 있는 최대 이익을 계산하는 프로그램을 작성하라. 예를 들어 입력 정수들이 [100, 180, 260, 310, 40, 535, 695]라고 하자. 이것은 첫째 날의 주식 가격은 100원, 둘째날은 180원, 셋째날은 260원, 그리고 마지막 날은 695원임을 의미한다. 최대의 이익을 얻기 위해서는 첫째 날에 주식을 사고 넷째 날에 팔고, 그리고 다시 5번째 날에 사서 7번째 날에 팔면 총 $(310-100) + (695-40) = 865$ 원의 이익을 얻을 수 있다.

입력 예(INPUT10.TXT)	출력
100 180 260 310 40 535 695 -1	865
10 9 8 7 6 5 4 3 2 1 -1	0
10 20 10 20 10 20 10 20 10 20 -1	50
2 4 4 4 8 8 8 5 3 10 8 6 -1	13

23. 키보드로 부터 연속해서 정수들을 입력받는다. 정수가 하나 씩 입력될 때 마다 현재까지 입력된 정수들을 오름차순으로 정렬하여 화면에 출력한다. 단, 새로 입력된 정수가 이미 배열에 저장되어 있다면 추가하는 대신 “duplicate”라고 출력한다. 사용자가 -1을 입력하면 프로그램을 종료한다.

입력 예	출력
5	5
2	2 5
5	duplicate
1	1 2 5
3	1 2 3 5
-1	

입력 예	출력
4	4
3	3 4
4	duplicate
45	3 4 45
12	3 4 12 45
5	3 4 5 12 45
4	duplicate
45	duplicate
2	2 3 4 5 12 45
1	1 2 3 4 5 12 45
7	1 2 3 4 5 7 12 45
-1	

24. 선택정렬(selection sort) 알고리즘은 다음과 같다. 배열 `data`에 `data[0]`에서 `data[n-1]`까지 n 개의 정수가 저장되어 있다. 먼저 `data[0] ~ data[n-1]` 사이의 정수들 중에서 가장 큰 정수를 찾는다. 그것을 `data[k]`라고 가정해보자. 그러면 `data[k]`와 `data[n-1]`을 교환(`swap`)한다. 이제 가장 큰 정수가 `data[n-1]`, 즉 맨 마지막 위치에 저장되었으므로 그 값에 대해서는 더 이상 생각할 필요가 없다. 이제 `data[0] ~ data[n-2]` 중에서 최대값을 찾는다. 그 값을 `data[p]`라고 하자. 그러면 다시 `data[p]`와 `data[n-2]`를 교환하고 `data[n-2]`에 대해서는 잊어버려도 된다. 이런 식으로 계속하면 마지막에는 `data[0]`와 `data[1]` 중에 최대값을 `data[1]`과 교환하면 전체의 정렬이 완료된다. 이 알고리즘을 구현하라. 입력은 먼저 정렬할 정수의 개수 n 이 주어지고 이어서 n 개의 정수들이 주어진다.

25. 사용자로부터 n 개의 정수를 입력받아 크기순으로 정렬한 후 중복된 수를 제거하는 프로그램을 작성하라. 입력 형식은 먼저 n 의 값이 주어지고 이어서 n 개의 정수들이 주어진다. 예를 들어 $n = 8$ 이고 입력된 정수들이 4, 7, 4, 12, 4, 10, 9, 7 이라면 중복을 제거하고 남은 정수들은 4, 7, 9, 10, 12 이다. 그러면 먼저 남은 정수의 개수 5를 출력하고 콜론(:)을 출력한 후 남은 정수들을 오름차순으로 출력한다.

입력 예	출력
8	
4 7 4 12 4 10 9 7	5: 4 7 9 10 12
5	
1 1 1 1 1	1: 1

26. 입력으로 n 개의 구간(interval)이 주어진다. 각 구간은 구간의 시작점과 끝점으로 표현된다. 이 구간들을 시작점이 빠른 순서대로 정렬하여 출력하는 프로그램을 작성하라. 시작점이 같은 경우 끝점이 빠른 것을 먼저 출력한다. 입력 형식은 먼저 n 의 값이 주어지고, 이어서 각 구간의 시작점과 끝점이 차례대로 주어진다. 각 구간의 시작점과 끝점은 정수이고, 끝점은 항상 시작점보다 크거나 같다.

입력 예	출력
8	
// 구간의 개수	
2 8	// 첫 번째 구간의 시작점과 끝점
4 12	// 두 번째 구간의 시작점과 끝점
4 7	
4 10	
4 12	
1 5	
9 11	
15 18	
6 8	// 마지막 구간의 시작점과 끝점

27. 수열에서 큰 값이 작은 값보다 앞서 나오는 경우 두 값을 역전된(inverted) 쌍이라고 부른다. 예를 들어 수열 4, 2, 1, 1, 3에는 (4, 2), (4, 1), (4, 1), (4, 3), (2, 1), (2, 1)의 총 6개의 역전된 쌍이 있다. 수열을 입력으로 받아서 역전된 쌍의 개수를 카운트하여 출력하는 프로그램을 작성하라. 키보드로부터 먼저 정수의 개수 N 을 입력받고, 이어서 N 개의 정수를 입력 받는다.

입력 예	출력
5 4 2 1 1 3	6
10 3 8 10 7 2 1 8 9 11 7	18
8 1 1 1 1 1 1 1 1	0
8 8 7 6 5 4 3 2 1	28

28. 입력으로 하나의 수열이 주어진다. 이 중 오름차순으로 정렬되어 있는 가장 긴 구간을 찾아서 그 구간의 길이를 출력하는 프로그램을 작성하라. 예를 들어 수열이 2, 3, -2, -2, -2, 7, 7, 9, 10, 10, 9, -3라면 밑줄 친 구간이 가장 긴 구간이고 그 길이는 8이다. 입력은 키보드로부터 받으며 먼저 수열의 길이 N 이 주어지고 이어서 N 개의 정수들이 주어진다.

입력 예	출력
12 2 3 -2 -2 -2 7 7 9 10 10 9 -3	8
10 3 7 1 8 0 2 6 2 1 9	3
12 9 3 7 2 4 6 12 8 3 2 9 1	4
12 9 9 9 9 8 9 10 3 4 5 1 2	4
6 1 1 1 1 1 1	6

29. 입력으로 N 개의 정수와 또 하나의 정수 K 가 주어진다. N 개의 정수들 중에서 합이 K 를 초과하지 않으면서 가장 긴 구간을 찾아 그 구간의 길이를 출력하는 프로그램을 작성하라. 키보드로부터 먼저 정수의 개수 N 을 입력받고, 이어서 N 개의 정수를 입력 받은 후, 마지막으로 정수 K 를 입력 받는다.

입력 예	출력
6 4 1 1 2 3 1	3
5 4 5 6 7 8	0
3 12 8 1 3 4 1 2 7 2 1 3 1 1	5
10 8 1 1 1 1 1 1 1 1	8
8	

30. 입력으로 N 개의 폐구간(closed interval)이 주어진다. 이중 겹치는 길이가 가장 긴 두 구간을 찾아서 출력하는 프로그램을 작성하라. 예를 들어 $N = 4$ 이고 입력된 구간들이 $[0, 3]$, $[3, 6]$, $[2, 7]$, $[4, 5]$ 라면 구간 $[3, 6]$ 과 $[2, 7]$ 이 겹치는 길이는 3이고 가장 길다. 입력은 우선 구간의 개수 N 이 주어지고 이어서 N 개의 구간의 시작점과 끝점이 주어진다. 각 구간의 끝점은 항상 시작점보다 크거나 같다. 두 구간을 출력할 때 반드시 입력에서 먼저 나온 구간을 먼저 출력해야 한다. 출력의 형식은 아래 예시를 따라라.

입력 예	출력
4 0 3 3 6 2 7 4 5	[3, 6] [2, 7]
8 1 8 2 7 12 37 23 28 34 48 5 12 8 29 4 13	[12, 37] [8, 29]
6 10 17 12 21 9 18 34 57 19 41 24 39	[19, 41] [24, 39]

31. 7개의 1~13 사이의 정수를 키보드로부터 입력받아 스트레이트(straight)가 성립하는지 판단하여 YES 혹은 NO라고 출력하는 프로그램을 작성하라. 스트레이트란 7개의 정수들 중 어떤 5개가 연속된 숫자인 경우를 말한다. 예를 들어서 입력된 정수가 5, 2, 5, 11, 4, 3, 6이면 이들 중 2, 3, 4, 5, 6이 있으므로 스트레이트가 성립한다.

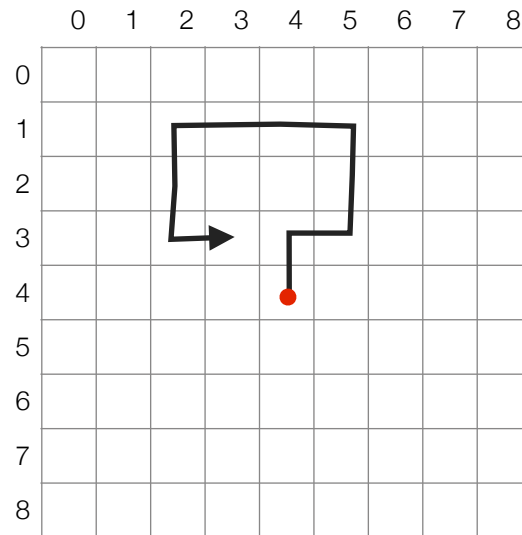
입력 예	출력
5 2 5 11 4 3 6	YES
8 8 12 5 5 3 4	NO
3 5 7 6 1 9 13	NO
9 10 11 8 2 2 7	YES

32. 0 혹은 1로 이루어진 길이가 N 인 수열이 주어진다. 이 중 하나 이하의 0을 1로 교체해서 만들 수 있는 가장 긴 연속된 1의 개수를 구하는 프로그램을 작성하라. 예를 들어 $N = 13$ 이고, 입력된 수열이 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1 라고 하자. 그러면 10번째 0을 1로 교체하면 연속된 1의 개수는 7개가 된다. 입력으로 먼저 N 의 값이 주어지고 이어서 N 개의 0 혹은 1이 주어진다.

입력 예	출력
13 1 1 0 0 1 0 1 1 1 0 1 1 1	7
5 1 1 1 1 0	5
4 1 1 1 1	4
4 0 0 0 0	1
12 0 1 0 1 1 0 1 1 1 0 1 1	6

33. [Puppy's Escape] 강아지가 $N \times N$ 크기의 2차원 배열의 가운데 위치 ($N/2, N/2$)에서 출발한다. N 은 홀수이다. 상, 하, 좌, 우 4방향으로 인접한 셀(cell)들 중에서 방문한 적이 없는 한 셀을 동일한 확률로

랜덤하게 선택하여 한 칸 이동한다. 가령 아래 그림에서 강아지의 현재 위치는 (3, 3)이고 이웃한 위치들 중에서 아직 방문하지 않은 위치는 (2, 3)과 (4, 3)으로 2곳이다. 따라서 두 위치중 하나를 1/2의 확률로 선택하여 이동한다. 배열의 가장자리 셀에 도착하면 탈출에 성공한 것이다. 하지만 아무 곳으로도 이동할 수 없는 상태에 처하면 탈출에 실패한 것이다. 아래 그림의 예에서 만약 위쪽 방향을 선택하여 (2, 3)으로 이동한다면 그 다음에는 어떻게 하더라도 탈출에 성공할 수 없다. 입력으로 하나의 홀수 $N \leq 100$ 을 받아서 강아지가 탈출에 성공할 확률을 시뮬레이션으로 계산하는 프로그램을 작성하라. 실험 횟수는 10,000번으로 하라. 정답은 없으며 아래의 예와 유사한 값이 출력되면 된다.



입력 예	출력 (정답이 없고 유사하면 됨)
5	1.0
9	0.966
21	0.644
51	0.117
71	0.033
99	0.005