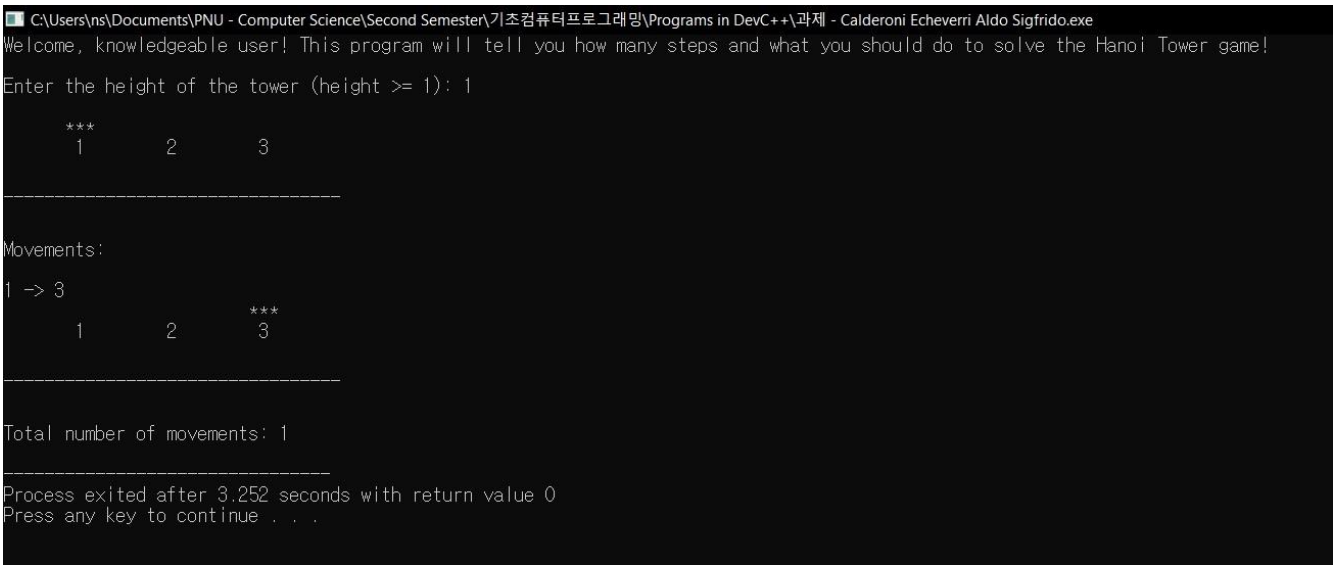
하노이 탑을 해결해 주고 단계를 보여 주는 프로그램

```
#include <stdio.h>

void print_hanoi_towers(int n, int towers[3][n]) {
	int tower_width = (n-1) * 2 + 3;
	int total_width = tower_width * 3 + 24;
	int i;

	for(i = 0; i < n; i++) {
		int a;
		for(a = 0; a < 3; a++) {
			int calc = (tower_width - (towers[a][i])) / 2;

			int j;
			for(j = 0; j < calc + 6; j++)
				printf(" ");
			if(towers[a][i] != 0) {
				for(j = 0; j < towers[a][i]; j++)
					printf("*");
				for(j = 0; j < calc; j++)
					printf(" ");
			}
			else {
				for(j = 0; j < calc + 1; j++)
					printf(" ");
			}
		}
		printf("\n");
	}

	for(i = 0; i < 6 + (tower_width / 2); i++)
		printf(" ");
	printf("1");
	for(i = 0; i < 6 + 2 * (tower_width / 2); i++)
		printf(" ");
	printf("2");
	for(i = 0; i < 6 + 2 * (tower_width / 2); i++)
		printf(" ");
	printf("3");

	printf("\n\n");
	for(i = 0; i < total_width; i++)
		printf("_");
	printf("\n\n");
}
```

```
int hanoi_tower(int n, int nc, int towers[3][nc], int ox, int tx, int mx) {
	static int counter = 0;
	counter++;

	if (n < 1) {
		printf("Error: n >= 1\n");
		return 0;
	}
	else if (n == 1) {
		int det1 = 0;
		while(towers[ox][det1] == 0)
			det1++;

		int det2 = 0;
		while(towers[tx][det2] == 0 && det2 != nc)
			det2++;

		towers[tx][det2 - 1] = towers[ox][det1];
		towers[ox][det1] = 0;
		printf("%d -> %d\n", ox + 1, tx + 1);
		print_hanoi_towers(nc, towers);
		return counter;
	}
	else {
		hanoi_tower(n-1, nc, towers, ox, mx, tx);

		int det1 = 0;
		while(towers[ox][det1] == 0)
			det1++;

		int det2 = 0;
		while(towers[tx][det2] == 0 && det2 != nc)
			det2++;

		towers[tx][det2 - 1] = towers[ox][det1];
		towers[ox][det1] = 0;
		printf("%d -> %d\n", ox + 1, tx + 1);
		print_hanoi_towers(nc, towers);

		hanoi_tower(n-1, nc, towers, mx, tx, ox);
		return counter;
	}
}

int main(void) {
	int n;
```

```
	printf("Welcome, knowledgeable user! This program will tell you how many steps and what you should do to solve the Hanoi Tower game!\n\n");
	printf("Enter the height of the tower (height >= 1): ");
	scanf("%d", &n);

	int towers[3][n];
	int i;
	for(i = 0; i < n; i++) {
		towers[0][i] = i * 2 + 3;
		towers[1][i] = 0;
		towers[2][i] = 0;
	}

	printf("\n");
	print_hanoi_towers(n, towers);
	printf("\nMovements:\n\n");
	int counter = hanoi_tower(n, n, towers, 0, 2, 1);
	printf("\nTotal number of movements: %d\n", counter);

	return 0;
}
```



```
C:\Users\ns\Documents\PNU - Computer Science\Second Semester\기초컴퓨터프로그래밍\Programs in DevC++\과제 - Calderoni Echeverri Aldo Sigfrido.exe
Welcome, knowledgeable user! This program will tell you how many steps and what you should do to solve the Hanoi Tower game!

Enter the height of the tower (height >= 1): 2

       ***
      *****
        1          2          3

----------------------------------------

Movements:

1 -> 2

      *****       ***
        1          2          3

----------------------------------------
1 -> 3

                  ***       *****
        1          2          3

----------------------------------------
2 -> 3

                             ***
                            *****
        1          2          3

----------------------------------------

Total number of movements: 3

--------------------------------
Process exited after 7.641 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\ns\Documents\PNU - Computer Science\Second Semester\기초컴퓨터프로그래밍\Programs in DevC++\과제 - Calderoni Echeverri Aldo Sigfrido.exe
Welcome, knowledgeable user! This program will tell you how many steps and what you should do to solve the Hanoi Tower game!

Enter the height of the tower (height >= 1): 3

        ***
       *****
      *******
         1           2           3

--------------------------------------------

Movements:

1 -> 3

       *****
      *******                     ***
         1           2           3

--------------------------------------------

1 -> 2


      *******       *****         ***
         1           2           3

--------------------------------------------

3 -> 2

                     ***
      *******       *****
         1           2           3

--------------------------------------------

1 -> 3

                     ***
                    *****       *******
         1           2           3

--------------------------------------------

2 -> 1


        ***         *****       *******
         1           2           3

--------------------------------------------

2 -> 3

                               *****
        ***                    *******
         1           2           3

--------------------------------------------

1 -> 3
                                 ***
                                *****
                               *******
         1           2           3

--------------------------------------------


Total number of movements: 7

--------------------------------
Process exited after 3.896 seconds with return value 0
Press any key to continue . . .
```