

11장 프로그래밍 실습_C 코드

[프로그래밍 실습 1]

```
#include <stdio.h>
#include <math.h>
#include <Windows.h>
#pragma warning (disable : 4996)
void main()
{
    double a, b, c;          // 방정식의 계수 선언
    double x, x1, x2;        // 방정식의 해를 위한 변수 선언
    // 방정식의 계수들을 입력받아서 해를 구하는 반복문
    while (1)
    {
        printf("a, b, c 값을 입력하라.\n");
        // 방정식의 계수인 a를 입력받음
        printf("a = ");
        scanf("%lf", &a);
        // 방정식의 계수인 b를 입력받음
        printf("b = ");
        scanf("%lf", &b);
        // 방정식의 계수인 c를 입력받음
        printf("c = ");
        scanf("%lf", &c);
        // 방정식의 계수인 a가 0인 경우:
        if (a == 0)
        {
            // 방정식의 계수인 b가 0인 경우:
            if (b == 0)
            {
                // 방정식의 계수인 c가 0인 경우 :
                if (c == 0)
                    printf("부정\n");
                // 방정식의 계수인 c가 0이 아닌 경우 :
                else
                    printf("불능\n");
            }
            // 방정식의 계수인 b가 0이 아닌 경우:
            else
            {
                x = c / b;
                printf("일차방정식. \n x=%g\n", x);
            }
        }
        // 방정식의 계수인 a가 0이 아닌 경우 :
        else
        {
            // 판별식 D가 0보다 큰 경우 실근 2개 존재
            if (b * b - 4 * a * c > 0)
            {
                x1 = (-b + sqrt(b * b - 4 * a * c)) / (2 * a);
                x2 = (-b - sqrt(b * b - 4 * a * c)) / (2 * a);
                printf("2개의 실근\n x1 = %g x2 = %g\n", x1, x2);
            }
            else
            {
                // 판별식 D가 0보다 작을 경우 실근 없음
            }
        }
    }
}
```

```

        // 판별식 D가 0인 경우 중근 1개 존재
        if (b * b - 4 * a * c == 0)
        {
            x = -b / (2 * a);
            printf("중근\n x=%lf\n", x);
        }
        // 판별식 D가 0보다 작은 경우 허근 존재
        else
            printf("허근\n");
    }
}
system("PAUSE");
}

```

[프로그래밍 실습 2]

```

#include <stdio.h>
#include<Windows.h>
#pragma warning (disable : 4996)

int fib_1(int);          // 함수 원형 정의

void main()
{
    int i;
    // 자연수를 입력받음
    printf("몇 번째 항을 구할까? 자연수를 입력하라. : ");
    scanf("%d", &i);
    // 입력된 자연수에 대한 피보나치 수열을 구하는 함수를 통해 결과 출력
    printf("피보나치 수열의 %d번째 항은 %d이다.\n", i, fib_1(i));
    system("PAUSE");
}

// 피보나치 수열을 계산하는 함수
int fib_1(int a)
{
    int fib;
    fib = 0;
    // a가 0보다 클 때 계속 진행
    if (a > 0)
    {
        if ((a == 1) || (a == 2))
            fib += 1;
        else
            // 재귀함수 호출
            fib = fib_1(a - 1) + fib_1(a - 2);
    }
    // a가 0이면 fib를 반환
    return fib;
}

```

[프로그래밍 실습 3]

```
#include <stdio.h>
#include<Windows.h>
#pragma warning (disable : 4996)

int G_func(int);          // 함수 원형 정의

void main()
{
    int i;
    //  $F(N+2)=2F(N)+F(N+1)$ 과 같은 점화 관계에 대해 구하려고 하는 항을 입력받음
    printf("구하려고 하는 항을 입력하라. : ");
    scanf("%d", &i);
    // 재귀함수인 G_func를 통해 입력된 i에 대해 F(i)를 계산해서 결과 출력
    printf("%d번째 항의 값 F(%d) = %d\n", i, i-1, G_func(i-1));
    system("PAUSE");
}

// 재귀함수로 정의된 점화 관계
int G_func(int n)
{
    // 항이 0이거나 0보다 클 때 계속 진행
    if (n >= 0)
    {
        // 항이 1 아니면 0일 때 1을 반환
        if (n <= 1)
            return 1;
        // 점화 관계  $2F(N)+F(N+1)$ 을 반환
        return 2 * G_func(n - 2) + G_func(n - 1);
    }
}
```

[프로그래밍 실습 4]

```
#include <stdio.h>
#include<Windows.h>
#pragma warning (disable : 4996)

int cnt = 0; // 이동 횟수에 이용되는 전역 변수
void moveHanoi(char a, char b, char c, int n);
void main()
{
    int n;
    printf("input Hanoi number (exit : 음수 ) : ");
    scanf("%d", &n);

    // 임의의 음수를 입력할 때까지 계속 원판 개수인 n에 대한 하노이 탑의 이동 출력
    while (n > 0)
    {
        // 입력된 원판 개수인 n에 대해 하노이 탑의 이동 출력
        moveHanoi('A', 'B', 'C', n);
        cnt = 0;
        printf("input Hanoi number (exit : 음수 ) : ");
        scanf("%d", &n);
    }
    system("PAUSE");
}
```

```
}
```

```
// 재귀적으로 정의된 하노이 탑 이동 함수
```

```
void moveHanoi(char from, char temp, char to, int n)
```

```
{
```

```
    if (n == 1)
```

```
    {
```

```
        ++cnt;
```

```
        printf("%5d: 막대 %c에서 막대 %c로 원판 %d를 이동\n", cnt, from, to, 1);
```

```
    }
```

```
    else
```

```
    {
```

```
        moveHanoi(from, to, temp, n - 1);
```

```
        ++cnt;
```

```
        printf("%5d: 막대 %c에서 막대 %c로 원판 %d를 이동\n", cnt, from, to, n);
```

```
        moveHanoi(temp, from, to, n - 1);
```

```
    }
```

```
}
```