

HANOI'S TOWER

구현 중점 사항

- typedef
- stack
- recursive
- module

typedef

```
typedef char Data;
```

자료형을 재정의 하여 사용한다.

->자료형을 변경하기가 쉽다.(reusable)

Stack ADT

```
typedef struct stack{  
    Data ring[SIZE];  
    int top;  
    char name;  
}stack;
```

```
void stackInit(stack*);  
int stackEmpty(stack*);  
Data pop(stack*);  
Data top(stack*);  
void push(stack*,Data ring);
```

Recursive

```
void HanoiSort(stack *a, stack *b, stack *c,int n){
```

-스택 a에서 b를 거쳐서 c로 n개를 옮긴다.

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

```
        inputTower(c , outputTower(a));
```

-종료조건, return 을 실행하는 것이 아니다.(a에서 꺼내어 c에 넣는다.)

```
    }
```

```
    else{
```

```
        HanoiSort(a , c , b , n-1);
```

-n-1개를 b로 옮기고

```
        HanoiSort(a , b , c , 1);
```

-a의 하나 남은 것을 c로 옮긴다.

```
        HanoiSort(b , a , c , n-1);
```

-b에 있는 n-1개를 a를 거쳐 c로

옮긴다.

```
    }
```

```
}
```

Module

```
Data outputTower(stack *target);
```

-pop()

```
void inputTower(stack *target, Data ring);
```

-push()

```
void HanoiPrint(stack*,stack*,stack*);
```

-stackPrint()