TOH

Tower of Hanoi is a mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
3. No disk may be placed on top of a smaller disk. Design a program for Tower of Hanoi using recursion.

No of Disk =3 and no. of rod = 3

Take an example for 2 disks :

Let rod 1 = &#39;A&#39;, rod 2 = &#39;B&#39;, rod 3 = &#39;C&#39;.

Step 1 : Shift first disk from &#39;A&#39; to &#39;C&#39;.

Step 2 : Shift second disk from &#39;A&#39; to &#39;B&#39;.

Step 3 : Shift first disk from &#39;C&#39; to &#39;B&#39;.

The pattern here is :

Top Disk moved from A to C

Top Disk moved from A to B

Top Disk moved from C to B

Input Format

3

Output Format

Top Disk moved from A to B

Top Disk moved from A to C

Top Disk moved from B to C

Top Disk moved from A to B

Top Disk moved from C to A

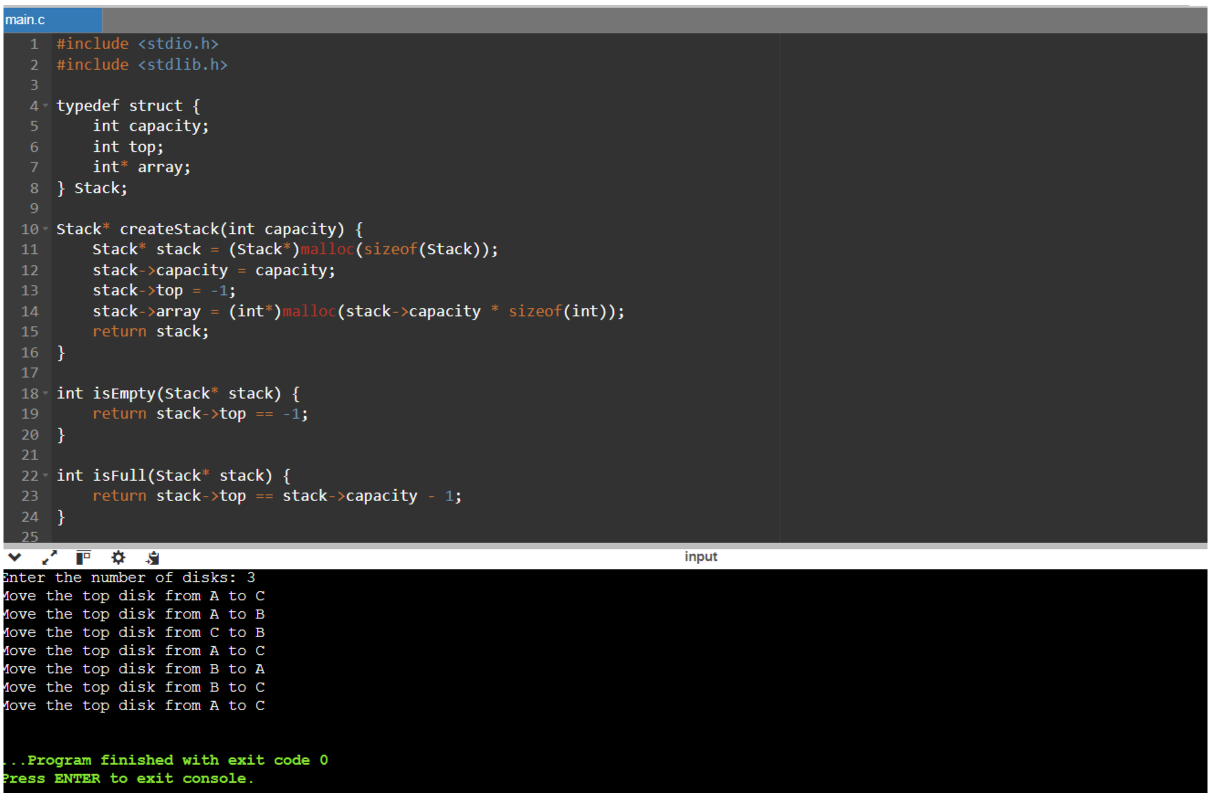
```
Top Disk moved from C to B
Top Disk moved from A to B
```

Code:

```
#include <stdio.h>
#include <stdlib.h>

typedef struct {
   int capacity;
   int top;
   int* array;
} Stack;

Stack* createStack(int capacity) {
   Stack* stack = (Stack*)malloc(sizeof(Stack));
   stack->capacity = capacity;
   stack->top = -1;
   stack->array = (int*)malloc(stack->capacity * sizeof(int));
   return stack;
}

int isEmpty(Stack* stack) {
   return stack->top == -1;
}

int isFull(Stack* stack) {
   return stack->top == stack->capacity - 1;
}

void push(Stack* stack, int item) {
   if (isFull(stack)) {
       printf("Stack overflow\n");
       return;
   }
   stack->array[++stack->top] = item;
}

int pop(Stack* stack) {
   if (isEmpty(stack)) {
       printf("Stack underflow\n");
       return -1;
   }
   return stack->array[stack->top--];
}

void towerOfHanoi(int n, char source, char auxiliary, char destination) {
   if (n == 1) {
       printf("Move the top disk from %c to %c\n", source, destination);
       return;
   }

   towerOfHanoi(n - 1, source, destination, auxiliary);
   printf("Move the top disk from %c to %c\n", source, destination);
   towerOfHanoi(n - 1, auxiliary, source, destination);
}

int main() {
   int n;
   printf("Enter the number of disks: ");
   scanf("%d", &n);
   towerOfHanoi(n, 'A', 'B', 'C');
   return 0;
```

}

Output:


main.c

```c
#include <stdio.h>
#include <stdlib.h>

typedef struct {
    int capacity;
    int top;
    int* array;
} Stack;

Stack* createStack(int capacity) {
    Stack* stack = (Stack*)malloc(sizeof(Stack));
    stack->capacity = capacity;
    stack->top = -1;
    stack->array = (int*)malloc(stack->capacity * sizeof(int));
    return stack;
}

int isEmpty(Stack* stack) {
    return stack->top == -1;
}

int isFull(Stack* stack) {
    return stack->top == stack->capacity - 1;
}

```

```
Enter the number of disks: 3
Move the top disk from A to C
Move the top disk from A to B
Move the top disk from C to B
Move the top disk from A to C
Move the top disk from B to A
Move the top disk from B to C
Move the top disk from A to C


...Program finished with exit code 0
Press ENTER to exit console.
```