Recursion And Pointer:

Recursion:

```c
//Call stack:
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#define testcase int tc; scanf("%d",&tc);while(tc--)
#define ll long long int
//Call stack
void world(){
    printf("Hello world\n");
}
void bangladesh(){
    printf("Hello Bangladesh\n");
    world(); // call world function...
    printf("Bangladesh\n");
}
void dhaka(){
    printf("Hello Dhaka\n");
    bangladesh(); // call world function...
    printf("Dhaka\n");
}
int main() { // main function at first call and call stack automatic call it.
    dhaka();
    printf("End.....!");

    return 0;
}
```

```c
Recursion: 1 2 3 4 5
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#define testcase int tc; scanf("%d",&tc);while(tc--)
#define ll long long int
int func(int i,int n){
    if(i==n+1) return 0;
    printf("%d\n",i);
    return func(i+1,n);
}
int main() {
    int n;
    scanf("%d",&n);
    func(1,n);
    return 0;
}
```

Recursion: 5 4 3 2 1

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#define testcase int tc; scanf("%d",&tc);while(tc--)
#define ll long long int
int func(int n){
    if(n==0) return 0; //just return..
    printf("\n%d",n);
    return func(n-1);
}
int main() {
    func(5);
    return 0;
}
```

Recursion: 5 4 3 2 1 Also.

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#define testcase int tc; scanf("%d",&tc);while(tc--)
#define ll long long int
void fun(int i){
    if(i==6) return;
    fun(i+1); // 5 4 3 2
    printf("%d\n",i);
}
int main() {
    fun(1);
    return 0;
}
```

Pointer:

Pointer:

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#define testcase int tc; scanf("%d",&tc);while(tc--)
#define ll long long int

int main() {
    int x = 10;
    int *ptr = &x; // 4 bytes store.
    /*
    printf("X er address: %p\n", &x); //%p
    printf("ptr er value: %p\n", ptr);
    printf("ptr er address: %p", &ptr);
    */
    x=200;
    printf("X er value: %d\n", x); //%p
    printf("x er dereferance value: %d\n", *ptr);

    return 0;
}
```

```
X er value: 200
x er dereferance value: 200
```

Call by value:

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#define testcase int tc; scanf("%d",&tc);while(tc--)
#define ll long long int
void fun(int x){
    printf("Fun x different address: %p\n",&x);
}
int main() {
    int x=10;
    printf("main x er address: %p\n",&x);
    fun(x);
    x = 200;
    return 0;
}
```

```
main x er address: 0x7ffe1d46717c
Fun x different address: 0x7ffe1d46715c
```

Call by reference:

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#define testcase int tc; scanf("%d",&tc);while(tc--)
#define ll long long int
void fun(int *p){ // x er address assign here
    printf("Main er x er value: %d\n",*p);
    *p=500; // dereferance...
}
int main() {
    int x=10;
    printf("x er address: %p\n", &x);
    fun(&x);
    printf("x er value: %d\n", x);
    return 0;
}
```

```
x er address: 0x7ffe7f4e280c
Main er x er value: 10
x er value: 500
```