```c
// C program to demonstrate use of * for pointers in C
#include <stdio.h>

// Types of operators - unary , binary , ternary

// Operators , Operand

// If single operand follows operator - Unary

// If operand operator operand , then ?

// Remember just * operand , is a unary operator

// where , always read * as "content of address"

// or "value in address"


// Always visualize , draw


// Similary unary & , always read as "address of"

// Why & in scanf and not in printf ?


// Referencing , dereferencing

int main()
{
    // A normal integer variable
    int var = 10;

    // A pointer variable that holds address of var.
    int *ptr = &var;

    // The output of this line may be different in different
    // runs even on same machine.
    printf("Address of var = %p\n", ptr);
    // Why ?

    // This line prints value at address stored in ptr.
    // Value stored is value of variable "var"
    printf("Value of var = %d\n", *ptr);

    // var is
    printf("var is %d\n", var);

    // We can also use ptr as lvalue (Left hand
    // side of assignment)
    *ptr = 20; // Value at address is now 20

    // This prints 20
    printf("After doing *ptr = 20, *ptr is %d\n", *ptr);

    // How about var ?
    printf("After doing *ptr = 20, var is %d\n", var);

    return 0;
}
```