

```
#include <stdio.h>

int main(){
    /*** The sizeof operator ***/
    // Tells us how many bytes does a variable take
    long longVar, longArr[10];
    printf("Size of this long variable is %d bytes \n", sizeof(longVar));
    // It even works for a datatype
    printf("Size of any char variable is %d byte\n", sizeof(char));
    // For arrays it prints the total size of the array in bytes
    printf("Total space required by this long array with 10 elements is %d bytes\n",
    sizeof(longArr));

    /*** Referencing a variable ***/
    // To get the location of a variable, simply use the & operator
    int i = 101;
    printf("This integer is stored at internal address %ld\n", &i); // Will print the address
    where i is stored

    /*** Creating and dereferencing pointers ***/
    // To access/modify the value at the location given by a pointer,
    // use the dereferencing operator *
    int *ptr = &i;
    printf("The integer value stored at a location pointed by ptr is %d\n", *ptr);
    *ptr = 20/5; // We can use *ptr just as we would have used i
    printf("The new value of i is %d\n", i);
    (*ptr)++; // We can use *ptr just as we would have used i
    printf("The newer value of i is %d\n", i);
    printf("It takes %d bytes to store a pointer\n", sizeof(ptr)); // pointer variables always
    take 8 bytes

    // Pointers in printf and scanf
    // Scanf requires the address where input is to be stored
    // So far we have been giving this address using &i
    // but we can just as well send a pointer which contains the address too
    scanf("%d", ptr);
    printf("The value input by the user is %d\n", i);
    printf("I can print value of i using ptr as well %d\n", *ptr);

    // NULL pointers and dangers
    int *qtr;
    qtr = NULL; // Never try to read or write to the NULL address
    printf("The NULL address is actually %ld\n", NULL);
    // The following line will cause code to crash - uncomment it and try
    // printf("The integer stored at NULL address is %d", *qtr);
    qtr = 10101020103123; // Should never hardcode addresses either
    // Internal addresses change everytime code is run
    // The following line will also cause code to crash - uncomment and try
    // printf("The integer stored at my address is %d", *qtr);
    // Only use addresses already allocated to program e.g. using &i.

    return 0;
}
```