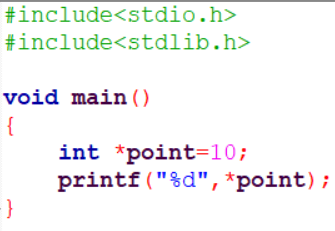
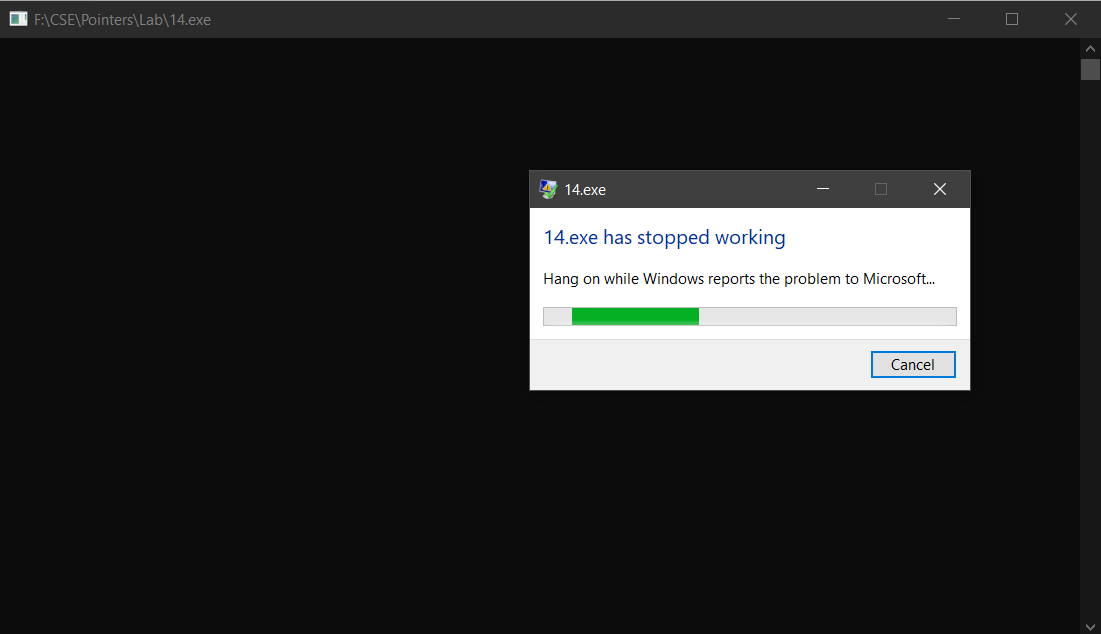
**C PROGRAMMING ASSIGNMENT**

S.ABHISHEK

**1.ACCESS VALUE WITHOUT ALLOCATING A MEMORY**





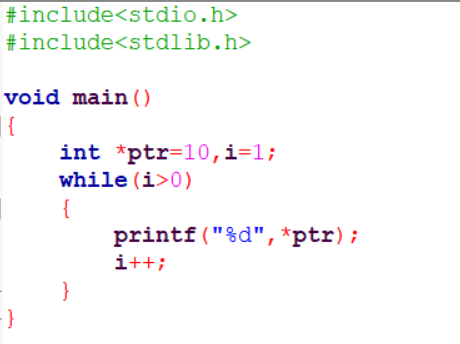
When we try to access the value without allocating a memory then it will give the Undefined Result.

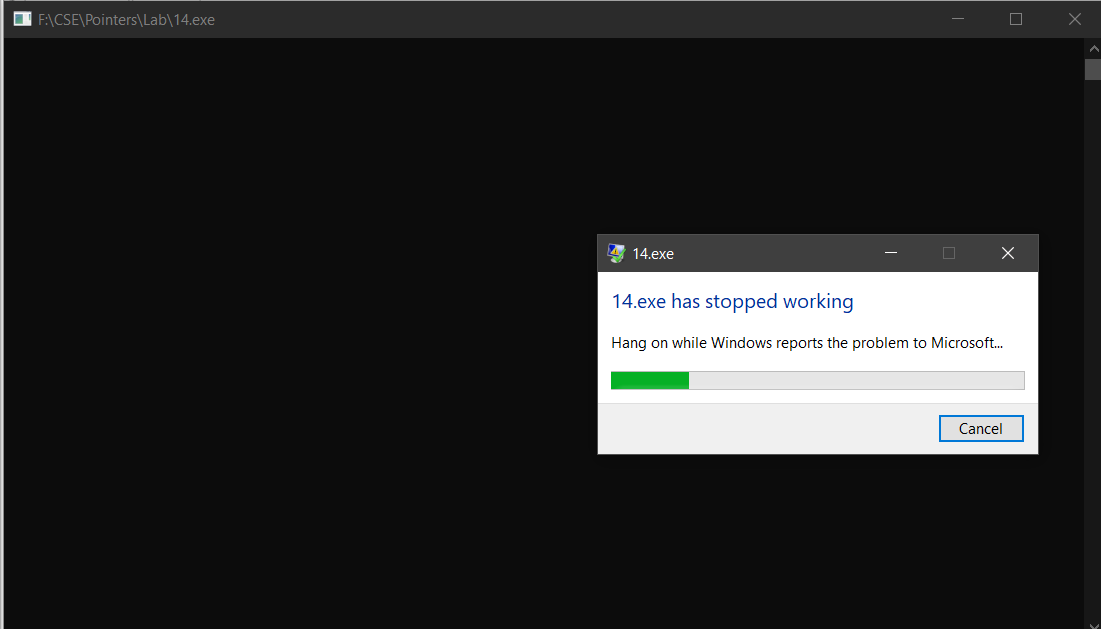
The program crashes while compiling and gets terminated.

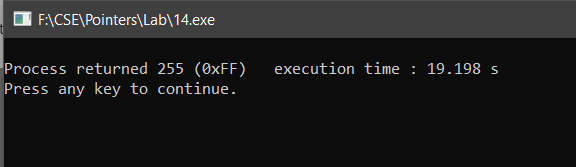
**ALLOCATING MEMORY WITHOUT FREEING**

Memory leaks happen when your program loses the address of some bit of dynamically allocated memory before giving it back to the operating system.

When this happens, your program can't delete the dynamically allocated memory, because it no longer knows where it is.





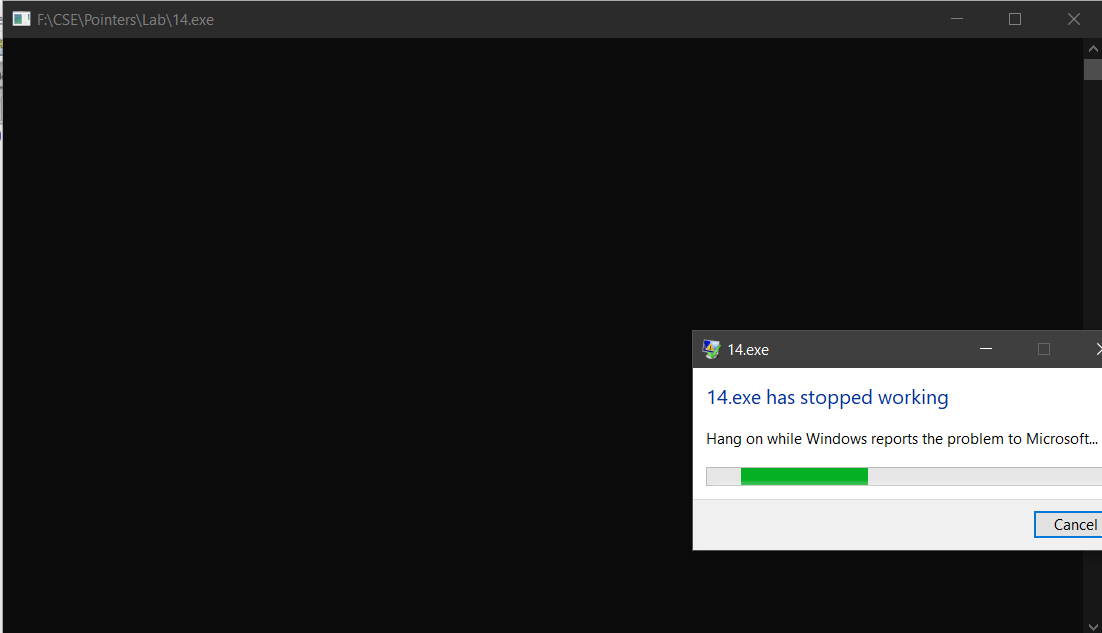


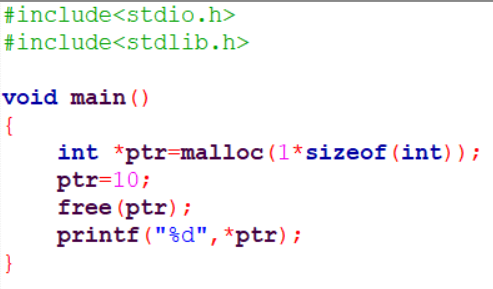
**2.ACCESS VALUE AFTER FREEING THE MEMORY**

When you call free (), the memory that points to is being freed.

But the value of the pointer stays untouched, making the pointer as a dangling pointer.

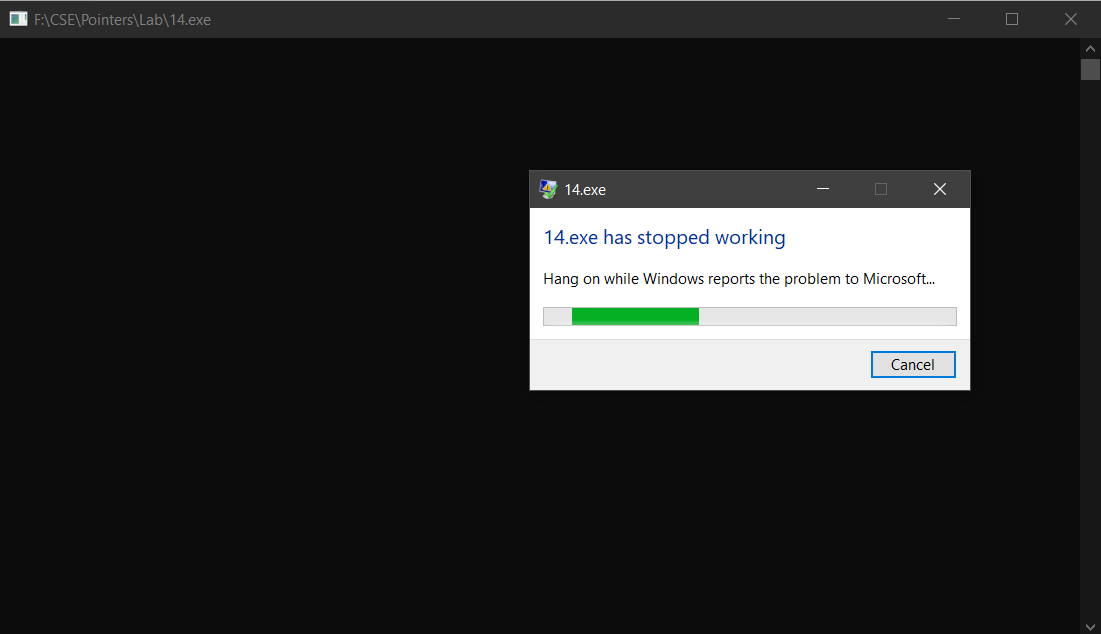
Accessing the memory that has already been freed produces **undefined behaviour**.





**3.ALLOCATING MEMORY WITHOUT FREEING**

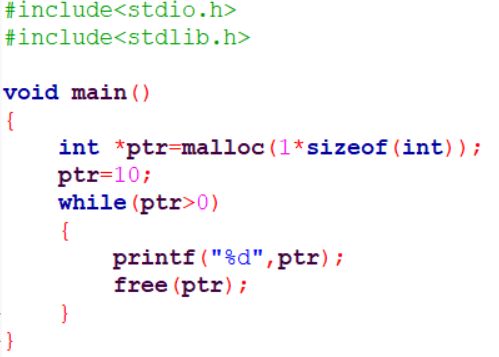
If we try to allocate memory without freeing it, there will be a memory loss and if we try to run an infinite loop there will be a program clash.

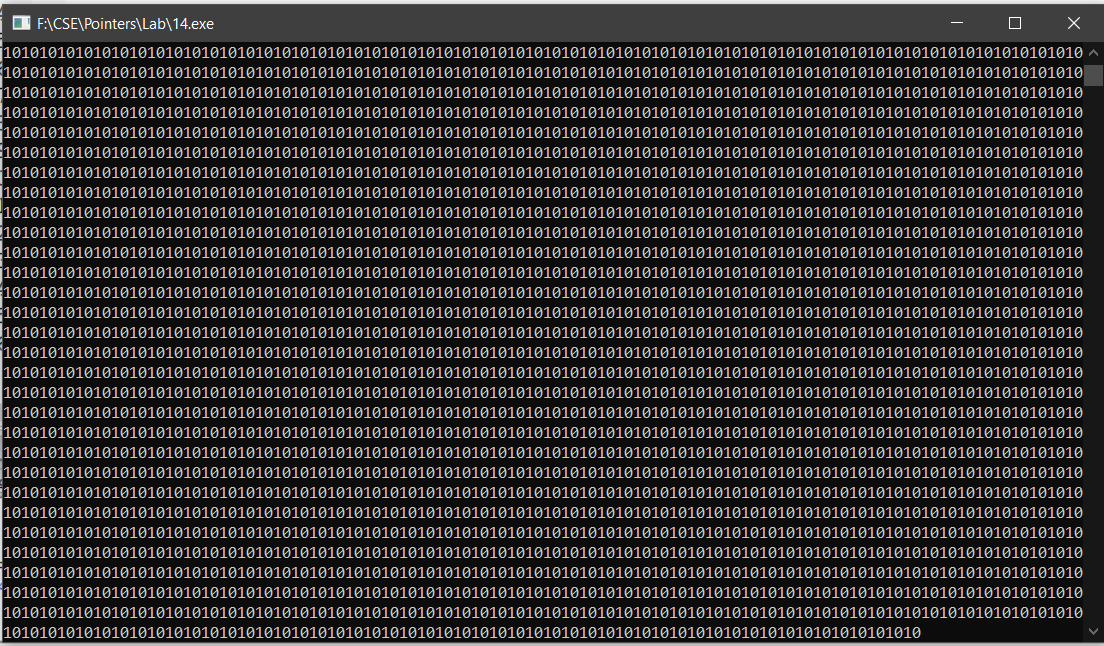


**4.ALLOCATING MEMORY WITH FREEING**

When we allocate an infinite loop with freeing the memory the loop keeps on running until CTRL+C is pressed.

The program runs forever without crashing.





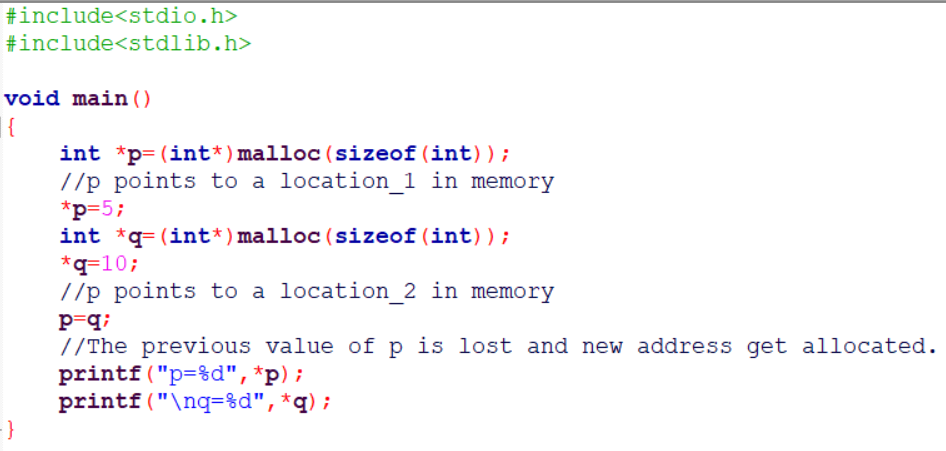
**5. REASSIGNING A PONTER TO ANOTHER LOCATION**

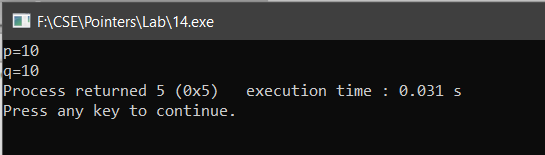
When a value assigned pointer is again reassigned to point another location the value which is already stored in the pointer gets deleted and the value in the reassigned location gets stored in it.

The access to the value 5 is lost. It is impossible to retrieve the value 5.

It can’t be freed either since the pointer is lost.

This leads to memory leak.



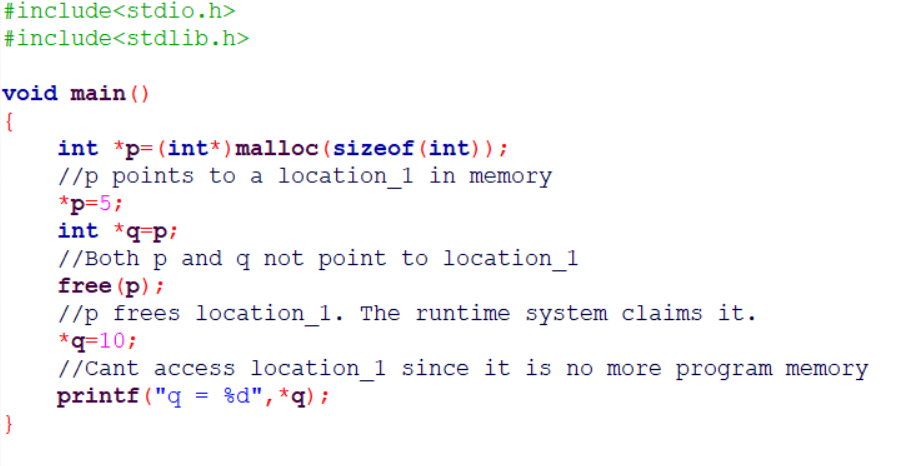


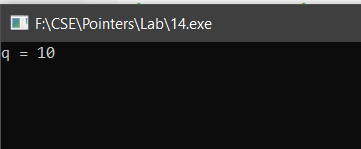
**6. FREEING THE POINTER WHEN ANOTHER POINTER IS ACCESSING THE SAME LOCATION**

There is a segmentation fault and the program gets crashed after printing the value of “q”.

This is because, the “q” tries to access the memory which is not the part of the program any more.

In short, q points to a location which does not legally belong to the program

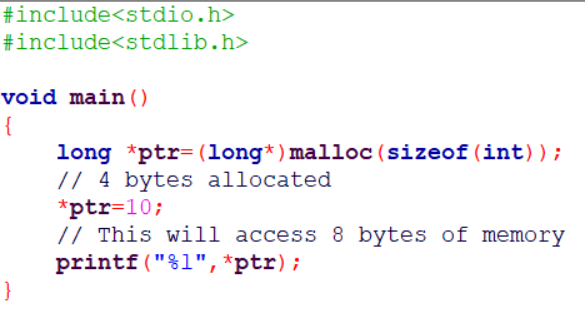


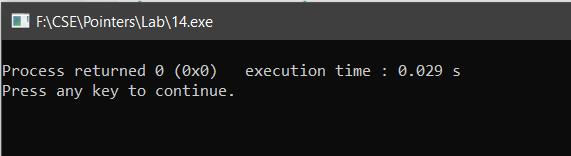


**7.INCORRECT TYPE CASTING**

Out of 8 bytes, only first 4 can be legally accessed. ii.

The runtime system will report an error when trying to assign value 10.





!.... Thankyou ….!