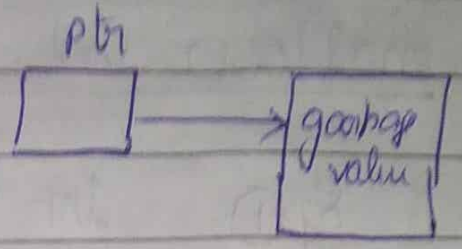
C_81 ⇒ Null Pointers in C

int *ptr;

↓

unitialized pointer.



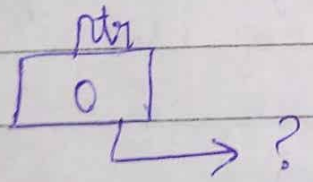
Not initialized.

* when we dereference this unitialized pointer without desired address; then it will show undefined behaviour.

* so better to initialize with NULL.

* NULL Pointer → Null itself is a pointer in C and corresponding NULL value is zero (NULL or 0).

int *ptr = NULL → It is predefined in header file.



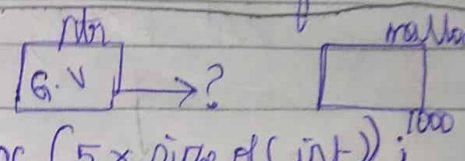
* It is a spuiall pointer which doesn't point to any valid address (or) does not refer to any valid address.

* Use of NULL Pointer :-

* If NULL pointer cannot dereference any address, then what is the use of it?

* We can use this NULL pointer to point to memory location which we allocate dynamically using malloc() and calloc() built in functions.

Eg:① int * ptr;

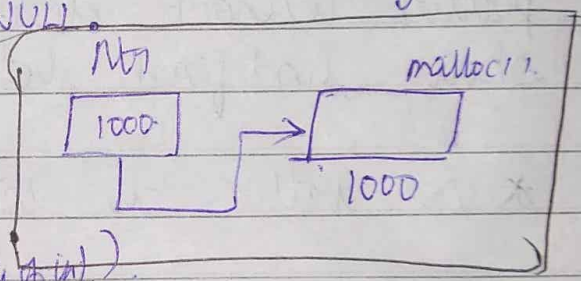


ptr = (int *) malloc (5 × size of (int));

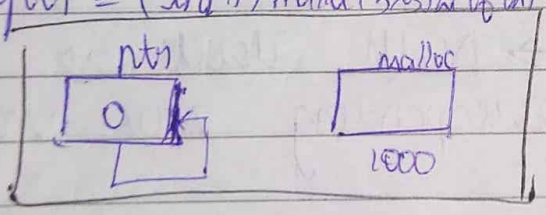
* Here we cannot dereffer.

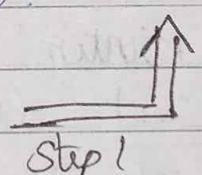
* (ii) When we allocate memory at runtime, we use NULL.

Eg② int *ptr = NULL;



ptr = (int *) malloc (5×size of int);





* Here we can dereffer the pointer ptr.

```
if (ptr == NULL)
{
  ---
}
else
{
  ---
}
```

Notes:

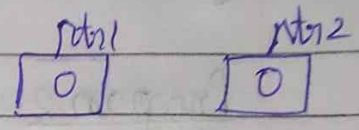
* NULL pointer does not refer to any valid object.

* We can get rid of undefined behaviour by initializing pointer to NULL rather than leaving it ~~with~~ uninitialized.

* We cannot dereference NULL pointer; if we do so then the program will crash.

Eg ③

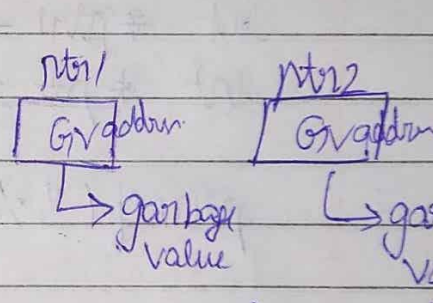
```
int *ptr1 = NULL;
int *ptr2 = NULL;
```



* Here both pointers are NULL values (i.e) 0 and same.

Eg ④

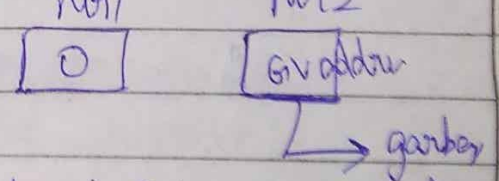
```
int *ptr1;
int *ptr2;
```



* Here both pointers are not initialized and their garbage value may be different.

Eg ⑤

```
int *ptr1 = NULL;
int *ptr2;
```



* Here ptr1 is NULL (0) but ptr2 is holding garbage value.

Program 1 :-

```
void main()
{
  int a=3;
  int *ptr1=NULL;
  int *ptr2;
  printf("%d\n", ptr1); ⇒ 0
```

↳ Here we do not dereference ptr1 instead we print value in ptr1.

```
  printf("%d\n", *ptr2); ⇒ garbage value
}
```

Program 2 :-

```
void main()
{
  int a=3;
  int *ptr1 = NULL;
  int *ptr2;

  printf("%d\n", *ptr1);
```

↳ Here when we try to dereference NULL pointer, then our program will crash.

(Null pointer.exe has stopped working)

```
}
```

Program 3 :-

```
Void main()
{
  int a=3;
  int *ptr1 = NULL;
  if(ptr1 == NULL)
  {
    printf("This is a null pointer");
  }
  else
  {
    printf("%d\n", *ptr1);
  }
```

* So, we can use this if else part, for error handling of NULL pointer.

Program 4 :

```
Void main()
{
  int a=3;
  int *ptr1 = NULL;
  int *ptr2 = NULL;
  if(ptr1 == ptr2)
  {
    printf("both are
                NULL pointers");
  }
  else { printf("%d\n", *ptr1); }.
}
```

Program 5 :-

```
void main()
{
  int a = 3;
  int *ptr1;    => g.v
  int *ptr2;    => g.v
```

} different

```
if (ptr1 == ptr2)
{
printf(" both are initialized");
}
else
{
  printf("%d\n", *ptr1);
}
}
```

↳ Here *ptr1 (when we dereference this initialized pointer), this program will crash.

Note :-

* Instead of NULL we can also write 0.

Eg :- int *ptr1 = 0;
int *ptr2 = 0;

PROBLEM 1:

```c
#include <stdio.h>
#include <stdlib.h>
/** NULL POINTERS **/
/** PROGRAM 1 **/
int main()
{
    int a=3;
    int *ptr1=NULL;
    int *ptr2;
    printf("%d\n",ptr1);
    //we cant dereference null pointer because it does not point to any address
    //But null pointer has value 0 stored in it instead of address
    printf("%d\n",*ptr2);
    //here ptr2 is not initialized with address & points to any garbage value
    getch();
}
```

```
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 6_JENNYS LECTURE_POIN
0

Process returned -1073741819 (0xC0000005)   execution time : 0.528 s
Press any key to continue.
```

PROBLEM 2:

```c
#include <stdio.h>
#include <stdlib.h>
/** NULL POINTERS **/
/** PROGRAM 2 **/
int main()
{
    int a=3;
    int *ptr1=NULL;
    printf("%d\n",*ptr1);
    //when we try to dereference null pointer, our program will crash
    getch();
}
```

```
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 6_JENNYS LECTURE_P

Process returned -1073741819 (0xC0000005)   execution time : 0.546 s
Press any key to continue.
```

PROBLEM 3:

```c
#include <stdio.h>
#include <stdlib.h>
/** NULL POINTERS **/
/** PROGRAM 3 **/
int main()
{
    int a=3;
    int *ptr1=NULL;
    if(ptr1==NULL)
    printf("It is a null pointer; pointing to nothing but holds value 0");
    else
    printf("%d\n",*ptr1);
    //we can use this if else part for error handling of null pointers
    getch();
}
```

"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 6_JENNYS LECTURE_POINTERS\

```
It is a null pointer; pointing to nothing but holds value 0
```

PROBLEM 4:

```c
#include <stdio.h>
#include <stdlib.h>
/** NULL POINTERS **/
/** PROGRAM 4 **/
int main()
{
    int a=3;
    int *ptr1=NULL;
    int *ptr2=NULL;
    if(ptr1==ptr2)
    {
     printf("ptr1 and ptr2 are null pointers\n");
     printf("ptr1 is null and holds value %d\n",ptr1);
     printf("ptr1 is null and holds value %d\n",ptr2);
    }
    else
    {
     printf("%d\n",*ptr1);
     printf("%d\n",*ptr2);
    }
    getch();
}
```

```
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 6_JENN
ptr1 and ptr2 are null pointers
ptr1 is null and holds value 0
ptr1 is null and holds value 0
```

PROBLEM 5:

```c
#include <stdio.h>
#include <stdlib.h>
/** NULL POINTERS **/
/** PROGRAM 5 **/
int main()
{
    int a=3;
    int *ptr1;
    int *ptr2;
    if(ptr1!=ptr2)
    {
     printf("ptr1 and ptr2 are not initialized\n");
     printf("Address in ptr1:%d\n",ptr1);
     printf("Address in ptr2:%d\n",ptr2);
    }
    else
    {
     printf("Value at address of ptr1:%d\n",*ptr1);
     printf("Value at address of ptr2:%d\n",*ptr2);
    }
    getch();
}
```

```
"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\
ptr1 and ptr2 are not initialized
Address in ptr1:10884144
Address in ptr2:156
```

PROBLEM 6:

```c
#include <stdio.h>
#include <stdlib.h>
/** NULL POINTERS **/
/** PROGRAM 6 **/
int main()
{
    int a=3;
    int *ptr1;
    int *ptr2;
    if(ptr1!=ptr2)
    {
     printf("ptr1 and ptr2 are not initialized\n");
     printf("Value at address of ptr1:%d\n",*ptr1);
     printf("Value at address of ptr2:%d\n",*ptr2);
     //we cant derefernce null pointer
    }
    else
    {
     printf("Address in ptr1:%d\n",ptr1);
     printf("Address in ptr2:%d\n",ptr2);
    }
    getch();
}
```

"D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 6_JENNYS LECTURE_POINTE

```
ptr1 and ptr2 are not initialized
Value at address of ptr1:7803745

Process returned -1073741819 (0xC0000005)   execution time : 0.530 s
Press any key to continue.
```