C_135 ⇒ Dynamic Memory Allocation using realloc()

* We are going to dynamically allocate memory and then according to our needs we reallocate the memory.

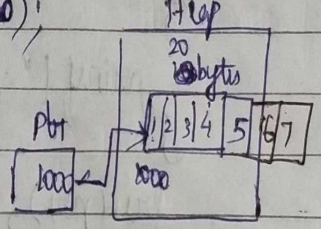
*Reallocate means we are going to resize the memory (increase / decrease) without loosing previous content.

* So it is responsible to resize the previously allocated memory using malloc/calloc.

Syntax:-

```
void* realloc(void *pointer, size_t size);
```

Eg: malloc(20);



↳ Consider we have allocated memory in heap section using malloc:

↳ And we have allocated 20 bytes of memory and the base address of allocated memory is returned to a pointer after typecasting the return type.

```
int *ptr;
ptr = (int*) malloc(5*sizeof(int));
```

↳ Now we want to add two more integers, so we can resize the memory from 20 bytes to 28 bytes (i.e) extra 8 bytes using realloc

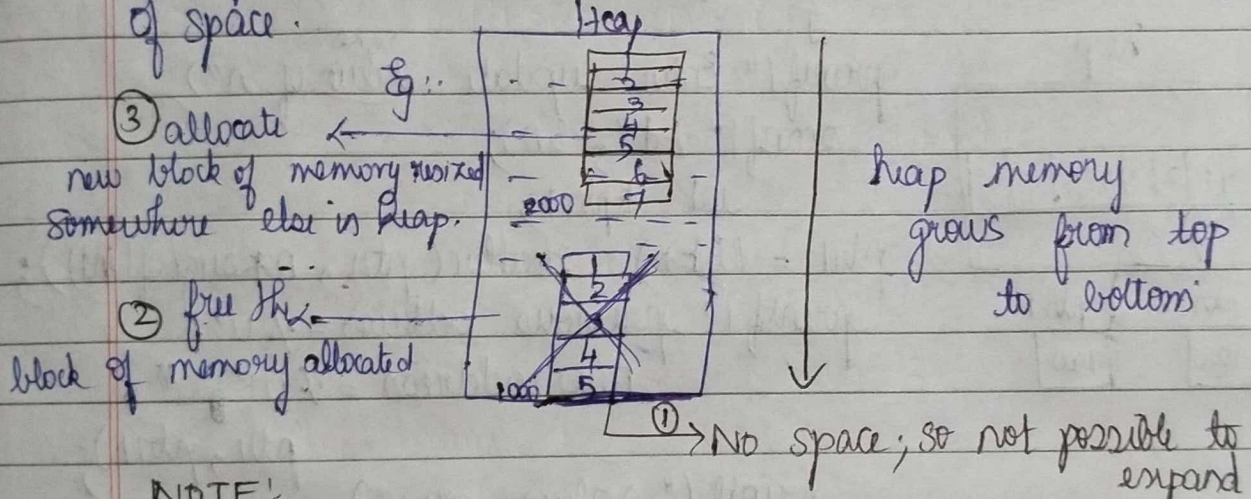
ptr1 (or) ptr = (int *) realloc (ptr, 7 * sizeof(int));

↳ Now the block is resized. We can store the base address of the memory allocated in same ptr or any other pointer variable (ptr1).

↳ Now the privious content of the memory & the base address of the memory is same; only chang is the memory size is extended.

NOTE:

* realloc will expand the memory size if it is possible to expand based on availability of space.



NOTE:

* If space is not available; then the previous allocated memory is deallocated and it will allocate a new block of memory with resized bytes and the content of the previously stored is same as it copies the content when allocating new block of resized memory

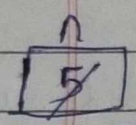
Finally it will return the new address and we can check it by printing the address.

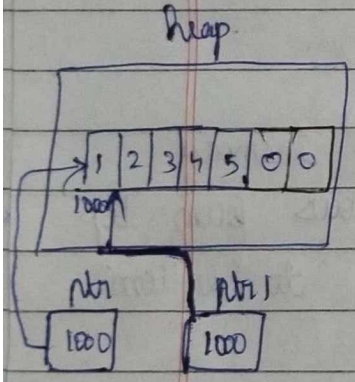
NOTE:

* If you decrease the allocated size then we will loose the content stored in the memory.

Program (1)

```
int *ptr, n, i;
printf("Enter value of n");
scanf("%d", &n);
ptr = (int*) calloc(n, sizeof(int));
printf("Enter values:");
for(i=0; i<n; i++)
{
    scanf("%d", (ptr+i));
}
printf("Enter update value of n");
scanf("%d", &n);
    int *ptr1;
ptr1 = (int*) realloc(ptr, n*sizeof(int));
printf("previous address = %u", ptr);
    new address = %u", ptr1);
printf("values are:");
for(i=0; i<n; i++)
{
    printf("%d", *(ptr1+i));
}
free(ptr1); free(ptr);
```





Assignment:

* Write a realloc program to decrease the previously allocated block of memory size.

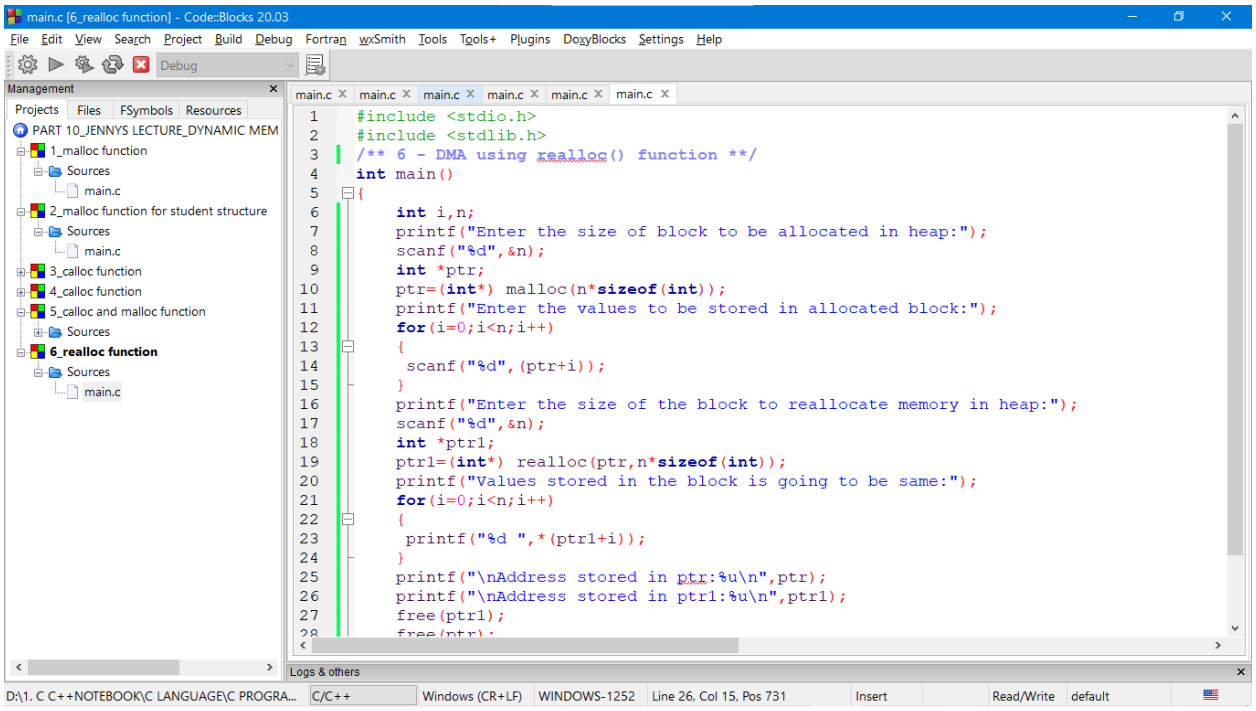
NOTE:

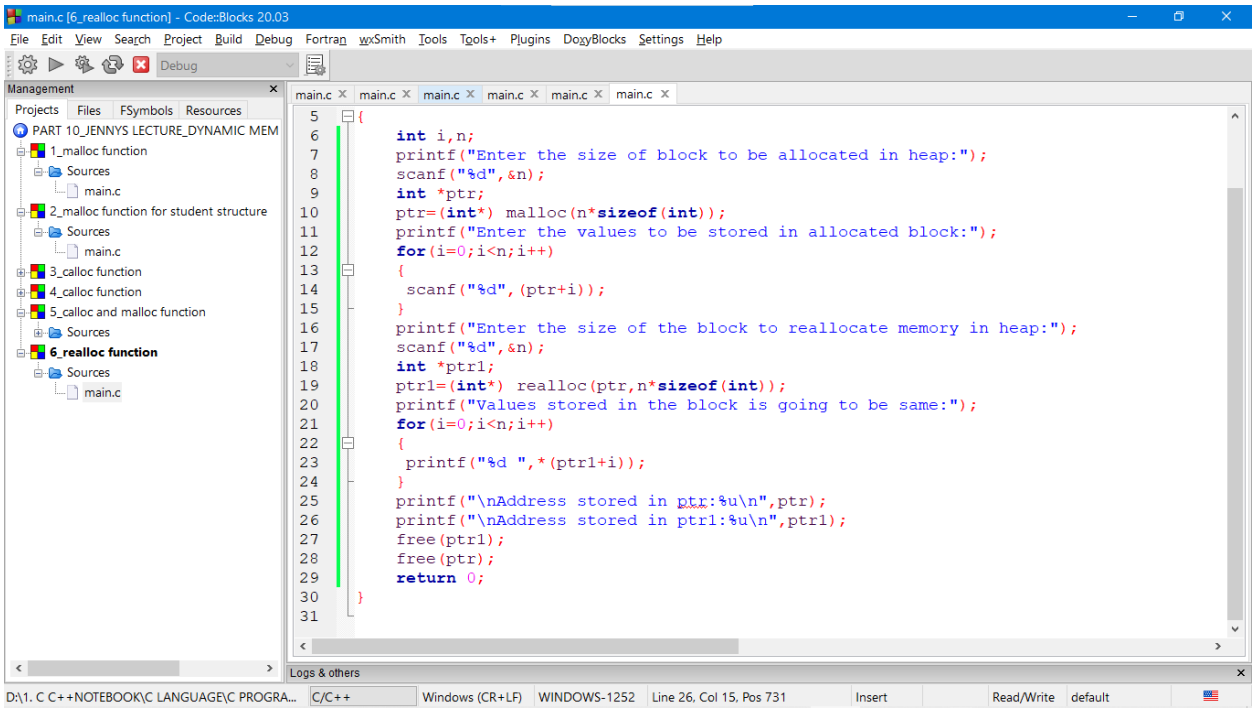
* Instead of specifying the previous pointer address if we specify NULL then the realloc will act as malloc (ie) will allocate a new block of memory and here we don't have any previous contents instead uninitialized memory space.

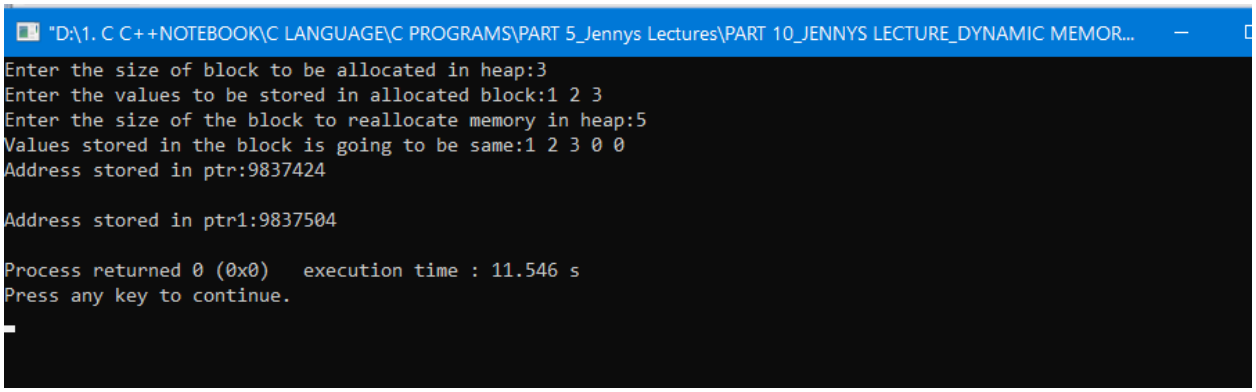
acts like malloc ← Eg: ptr1 = (int *) realloc (NULL, n * sizeof(int));

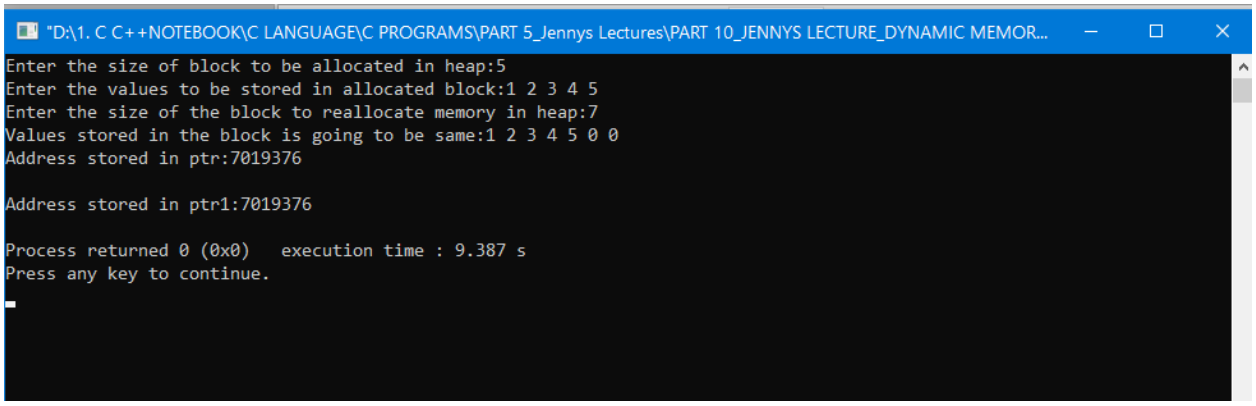
If we pass NULL instead of size of memory allocation, then it will act as free pointer (ie) we don't specify memory size but we pass the pointer address and NULL size then this will free the pointer pointing to this address.

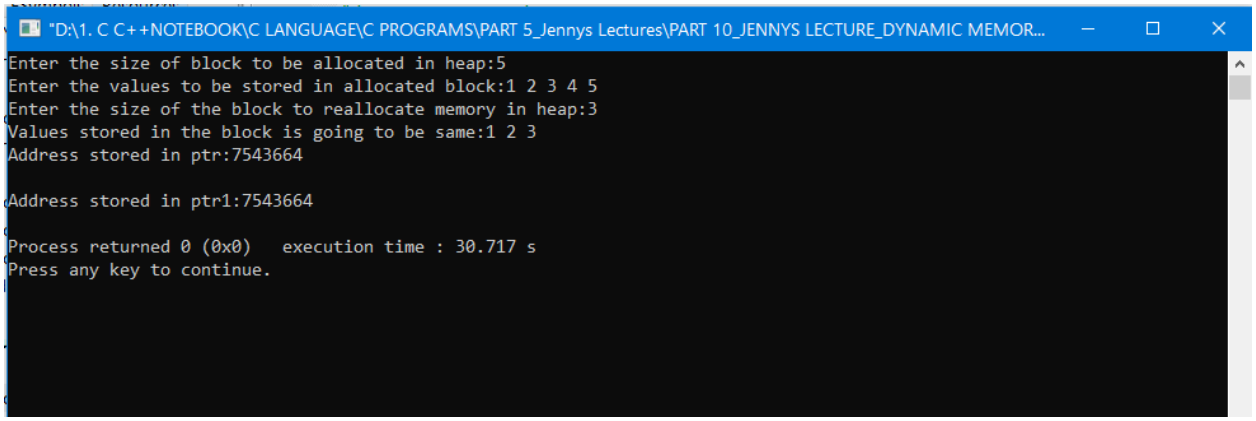
acts as free pointer ← Eg: ptr1 = (int *) realloc (ptr, NULL);

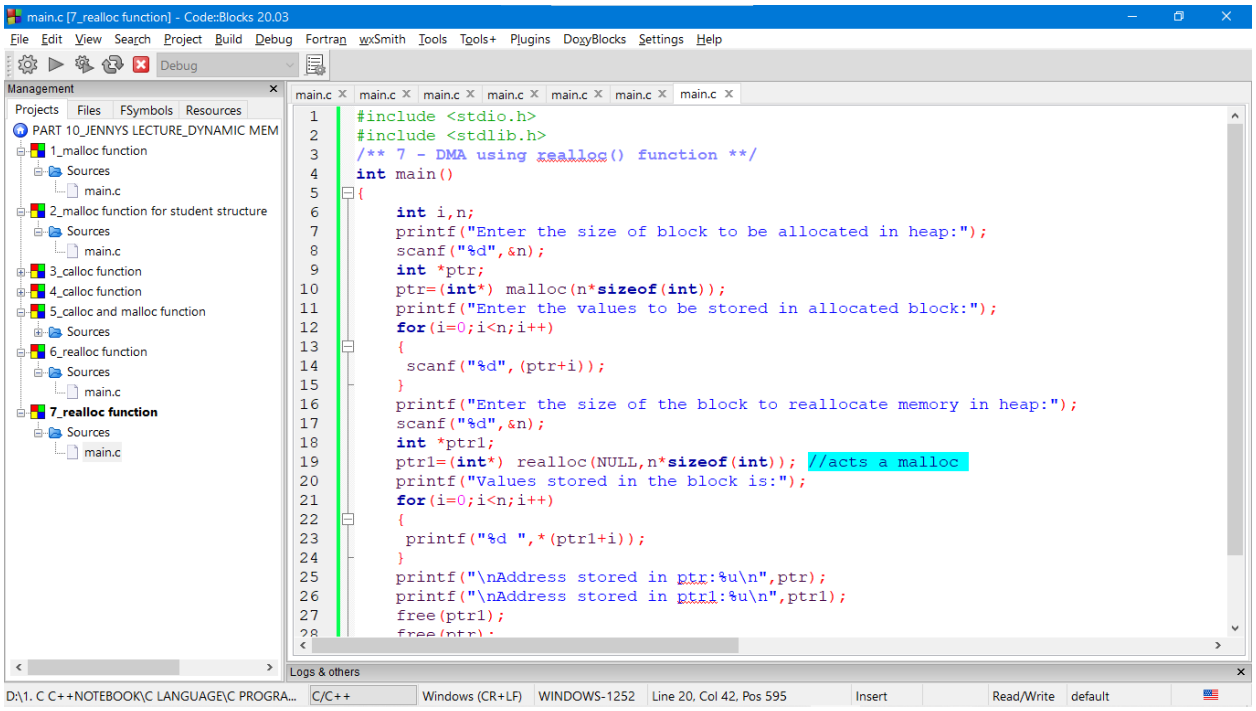

main.c [6_realloc function] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
Management
Projects Files FSymbols Resources
PART 10_JENNYS LECTURE_DYNAMIC MEM
1_malloc function
Sources
main.c
2_malloc function for student structure
Sources
main.c
3_calloc function
4_calloc function
5_calloc and malloc function
Sources
6_realloc function
Sources
main.c
main.c main.c main.c main.c main.c main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 /** 6 - DMA using realloc() function **/
4 int main()
5 {
6 int i,n;
7 printf("Enter the size of block to be allocated in heap:");
8 scanf("%d",&n);
9 int *ptr;
10 ptr=(int*) malloc(n*sizeof(int));
11 printf("Enter the values to be stored in allocated block:");
12 for(i=0;i<n;i++)
13 {
14 scanf("%d",(ptr+i));
15 }
16 printf("Enter the size of the block to reallocate memory in heap:");
17 scanf("%d",&n);
18 int *ptr1;
19 ptr1=(int*) realloc(ptr,n*sizeof(int));
20 printf("Values stored in the block is going to be same:");
21 for(i=0;i<n;i++)
22 {
23 printf("%d ",*(ptr1+i));
24 }
25 printf("\nAddress stored in ptr:%u\n",ptr);
26 printf("\nAddress stored in ptr1:%u\n",ptr1);
27 free(ptr1);
Logs & others
D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRA... C/C++ Windows (CR+LF) WINDOWS-1252 Line 26, Col 15, Pos 731 Insert Read/Write default

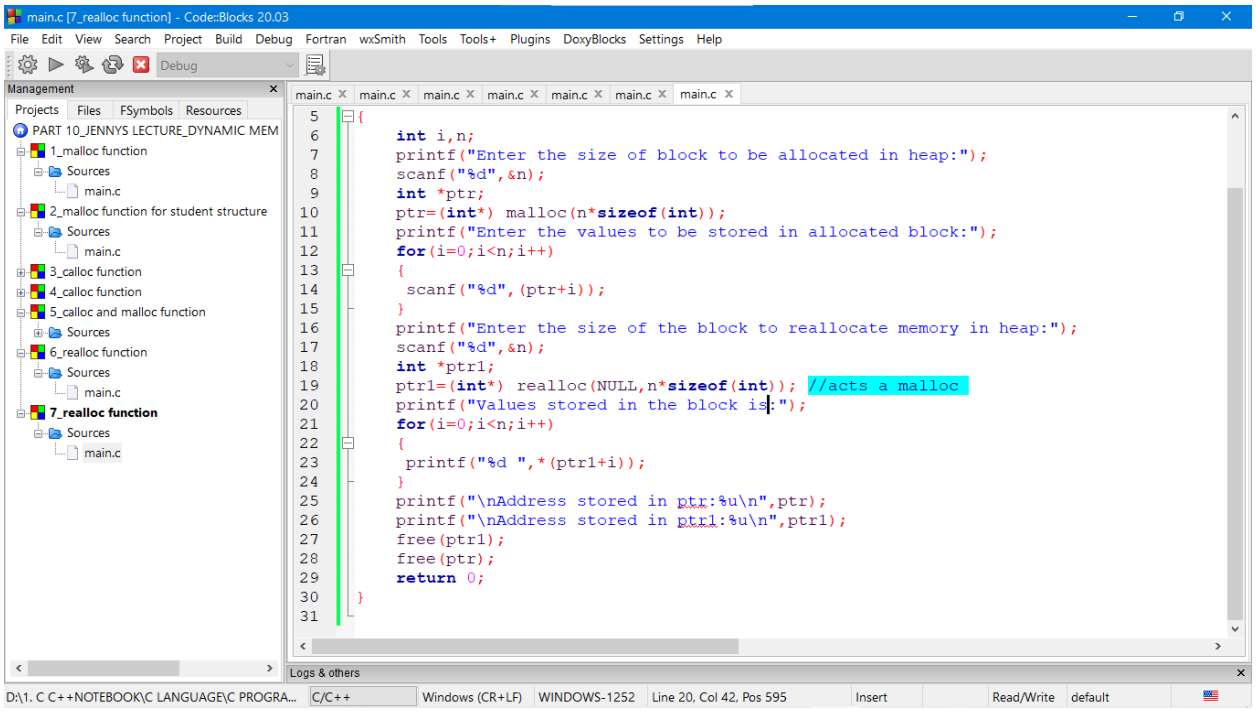

main.c [6_realloc function] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
Management
Projects Files FSymbols Resources
PART 10_JENNYS LECTURE_DYNAMIC MEM
1_malloc function
Sources
main.c
2_malloc function for student structure
Sources
main.c
3_calloc function
4_calloc function
5_calloc and malloc function
Sources
6_realloc function
Sources
main.c
main.c main.c main.c main.c main.c main.c
5 {
6 int i,n;
7 printf("Enter the size of block to be allocated in heap:");
8 scanf("%d",&n);
9 int *ptr;
10 ptr=(int*) malloc(n*sizeof(int));
11 printf("Enter the values to be stored in allocated block:");
12 for(i=0;i<n;i++)
13 {
14 scanf("%d",(ptr+i));
15 }
16 printf("Enter the size of the block to reallocate memory in heap:");
17 scanf("%d",&n);
18 int *ptr1;
19 ptr1=(int*) realloc(ptr,n*sizeof(int));
20 printf("Values stored in the block is going to be same:");
21 for(i=0;i<n;i++)
22 {
23 printf("%d ",*(ptr1+i));
24 }
25 printf("\nAddress stored in ptr:%u\n",ptr);
26 printf("\nAddress stored in ptr1:%u\n",ptr1);
27 free(ptr1);
28 free(ptr);
29 return 0;
30 }
31
Logs & others
D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRA... C/C++ Windows (CR+LF) WINDOWS-1252 Line 26, Col 15, Pos 731 Insert Read/Write default

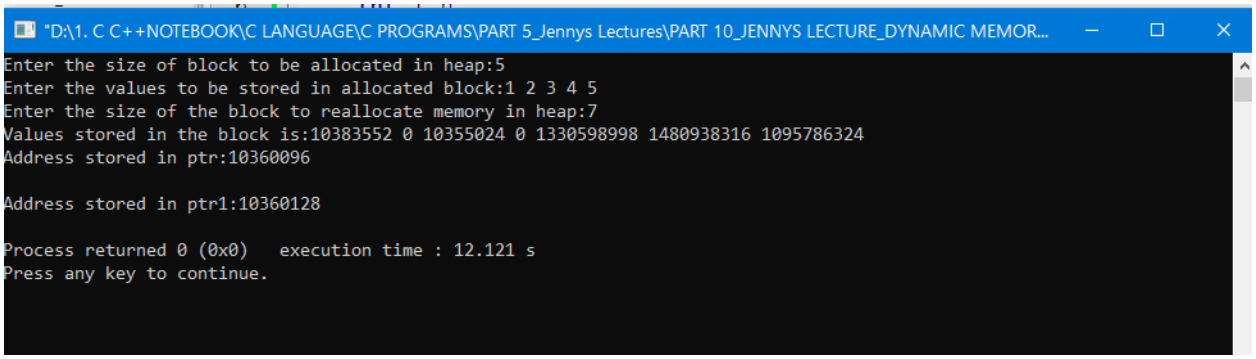

"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMOR...
Enter the size of block to be allocated in heap:3
Enter the values to be stored in allocated block:1 2 3
Enter the size of the block to reallocate memory in heap:5
Values stored in the block is going to be same:1 2 3 0 0
Address stored in ptr:9837424
Address stored in ptr1:9837504
Process returned 0 (0x0) execution time : 11.546 s
Press any key to continue.

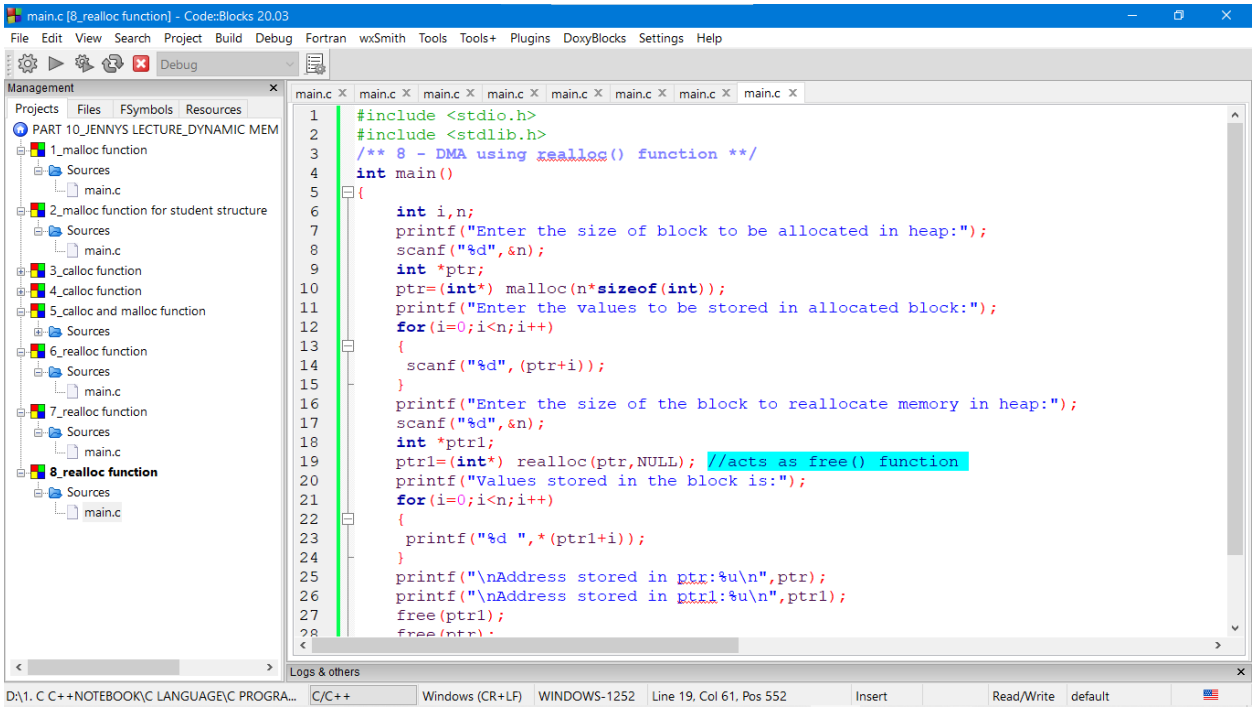

"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMOR...
Enter the size of block to be allocated in heap:5
Enter the values to be stored in allocated block:1 2 3 4 5
Enter the size of the block to reallocate memory in heap:7
Values stored in the block is going to be same:1 2 3 4 5 0 0
Address stored in ptr:7019376
Address stored in ptr1:7019376
Process returned 0 (0x0) execution time : 9.387 s
Press any key to continue.

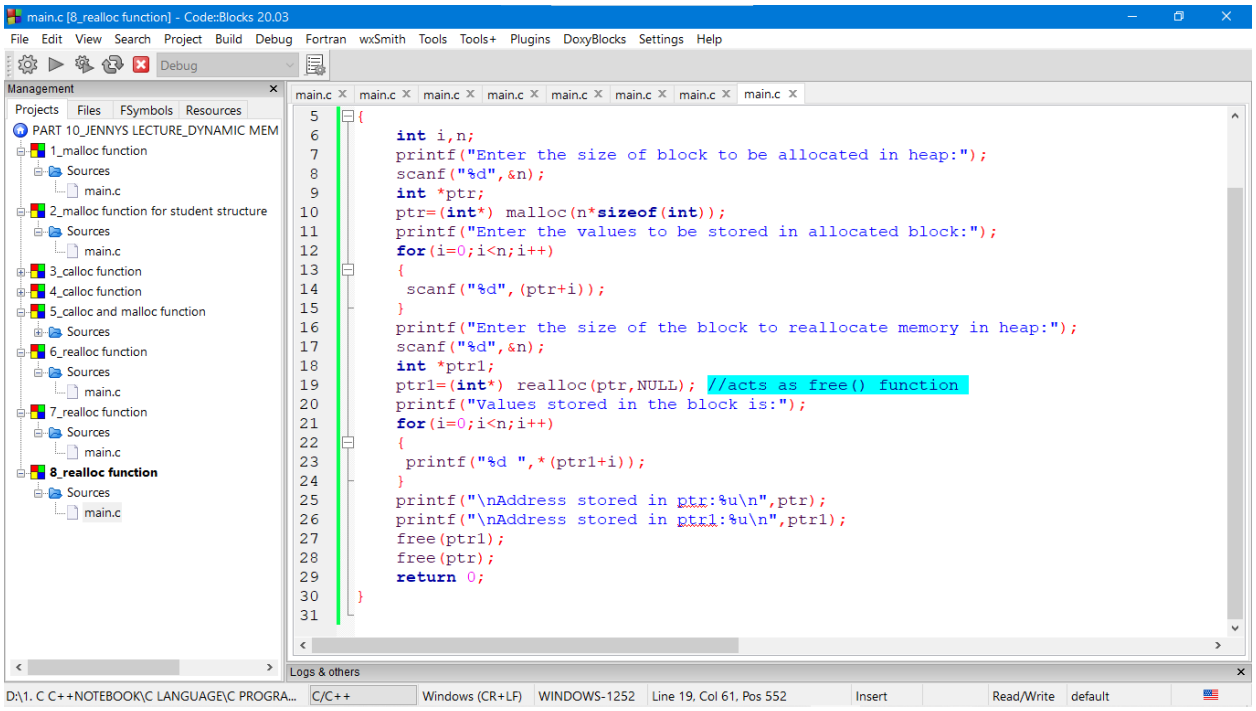

"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMOR...
Enter the size of block to be allocated in heap:5
Enter the values to be stored in allocated block:1 2 3 4 5
Enter the size of the block to reallocate memory in heap:3
Values stored in the block is going to be same:1 2 3
Address stored in ptr:7543664
Address stored in ptr1:7543664
Process returned 0 (0x0) execution time : 30.717 s
Press any key to continue.


main.c [7_realloc function] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
Management
Projects Files FSymbols Resources
PART 10_JENNYS LECTURE_DYNAMIC MEM
1_malloc function
Sources
main.c
2_malloc function for student structure
Sources
main.c
3_calloc function
4_calloc function
5_calloc and malloc function
Sources
6_realloc function
Sources
main.c
7_realloc function
Sources
main.c
main.c main.c main.c main.c main.c main.c main.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 /** 7 - DMA using realloc() function **/
4 int main()
5 {
6 int i,n;
7 printf("Enter the size of block to be allocated in heap:");
8 scanf("%d",&n);
9 int *ptr;
10 ptr=(int*) malloc(n*sizeof(int));
11 printf("Enter the values to be stored in allocated block:");
12 for(i=0;i<n;i++)
13 {
14 scanf("%d",(ptr+i));
15 }
16 printf("Enter the size of the block to reallocate memory in heap:");
17 scanf("%d",&n);
18 int *ptr1;
19 ptr1=(int*) realloc(NULL,n*sizeof(int)); //acts a malloc
20 printf("Values stored in the block is:");
21 for(i=0;i<n;i++)
22 {
23 printf("%d ",*(ptr1+i));
24 }
25 printf("\nAddress stored in ptr:%u\n",ptr);
26 printf("\nAddress stored in ptr1:%u\n",ptr1);
27 free(ptr1);
Logs & others
D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRA... C/C++ Windows (CR+LF) WINDOWS-1252 Line 20, Col 42, Pos 595 Insert Read/Write default


main.c [7_realloc function] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
Management
Projects Files FSymbols Resources
PART 10_JENNYS LECTURE_DYNAMIC MEM
1_malloc function
Sources
main.c
2_malloc function for student structure
Sources
main.c
3_calloc function
4_calloc function
5_calloc and malloc function
Sources
6_realloc function
Sources
main.c
7_realloc function
Sources
main.c
main.c main.c main.c main.c main.c main.c main.c
5 {
6 int i,n;
7 printf("Enter the size of block to be allocated in heap:");
8 scanf("%d",&n);
9 int *ptr;
10 ptr=(int*) malloc(n*sizeof(int));
11 printf("Enter the values to be stored in allocated block:");
12 for(i=0;i<n;i++)
13 {
14 scanf("%d",(ptr+i));
15 }
16 printf("Enter the size of the block to reallocate memory in heap:");
17 scanf("%d",&n);
18 int *ptr1;
19 ptr1=(int*) realloc(NULL,n*sizeof(int)); //acts a malloc
20 printf("Values stored in the block is:");
21 for(i=0;i<n;i++)
22 {
23 printf("%d ",*(ptr1+i));
24 }
25 printf("\nAddress stored in ptr:%u\n",ptr);
26 printf("\nAddress stored in ptr1:%u\n",ptr1);
27 free(ptr1);
28 free(ptr);
29 return 0;
30 }
31
Logs & others
D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRA... C/C++ Windows (CR+LF) WINDOWS-1252 Line 20, Col 42, Pos 595 Insert Read/Write default


"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMOR...
Enter the size of block to be allocated in heap:5
Enter the values to be stored in allocated block:1 2 3 4 5
Enter the size of the block to reallocate memory in heap:7
Values stored in the block is:10383552 0 10355024 0 1330598998 1480938316 1095786324
Address stored in ptr:10360096
Address stored in ptr1:10360128
Process returned 0 (0x0) execution time : 12.121 s
Press any key to continue.


main.c [8_realloc function] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
Management
Projects Files FSymbols Resources
PART 10_JENNYS LECTURE_DYNAMIC MEM
1_malloc function
Sources
main.c
2_malloc function for student structure
Sources
main.c
3_calloc function
4_calloc function
5_calloc and malloc function
Sources
6_realloc function
Sources
main.c
7_realloc function
Sources
main.c
8_realloc function
Sources
main.c
main.c main.c main.c main.c main.c main.c main.c main.c
#include <stdio.h>
#include <stdlib.h>
/** 8 - DMA using realloc() function **/
int main()
{
int i,n;
printf("Enter the size of block to be allocated in heap:");
scanf("%d",&n);
int *ptr;
ptr=(int*) malloc(n*sizeof(int));
printf("Enter the values to be stored in allocated block:");
for(i=0;i<n;i++)
{
scanf("%d",(ptr+i));
}
printf("Enter the size of the block to reallocate memory in heap:");
scanf("%d",&n);
int *ptr1;
ptr1=(int*) realloc(ptr,NULL); //acts as free() function
printf("Values stored in the block is:");
for(i=0;i<n;i++)
{
printf("%d ",*(ptr1+i));
}
printf("\nAddress stored in ptr:%u\n",ptr);
printf("\nAddress stored in ptr1:%u\n",ptr1);
free(ptr1);
Logs & others
D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRA... C/C++ Windows (CR+LF) WINDOWS-1252 Line 19, Col 61, Pos 552 Insert Read/Write default



Pointer ptr is free and hence when we try to access the pointer ptr which is now not pointing to any memory location will return a undefined behavior.