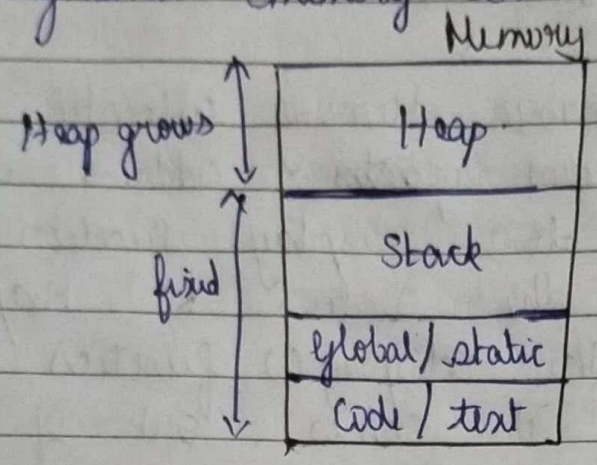
C_137 ⇒ Memory leak in C

(In proper use of heap; allocating memory dynamically but not releasing)

* It is a problem in both C/C++. This problem occurs due to improper use of dynamic memory or heap section.



→ Memory layout for any applications

* Heap section memory grow according to our needs and it is not fixed.

* But stack section is fixed based on the number of local variables and functions we declare

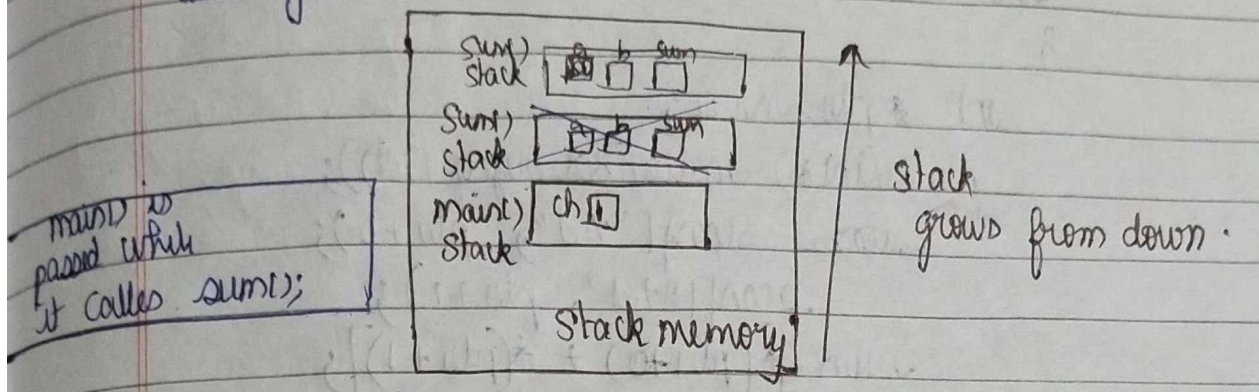
Program ① [This program is concept of memory allocation with SMM and no memory leak]

```
int sum()
{
    int sum=0;
    int a=10, b=11;
      sum = a+b;
    printf("%d", sum);
}
int main()
{
    int ch=1;
    while(ch);
    {
        sum();
        printf("Continue ?");
        scanf("%d", &ch);
    }
}
```

Ch value
1 → Continue
0 → Stop

* Now the program starts execution from main memory.

* So first main() will get allocated memory space in stack; it passes this when it calls sum() a separate stack frame is created for sum() and gets executed.



* Now when sum() gets executed and comes out of its scope; the memory allocated for this sum() stack frame is released or de-allocated.

* Now in main(), it gets unpassed since it needs to execute the next statement declared in it after sum() (i.e) printf("Continue") and enter the choice (scanf("%d", &ch);

* If the user gives choices as 1, the while loop continues and again a new stack frame for sum() is allocated.

* This process continues until we give choice as 0. This entire process is based on static memory allocation.

Date
Page

using DMA

Program ② :- [This program will free the heap section; no memory leak]

* Now if the same program; we allocate memory using DMA in heap section.

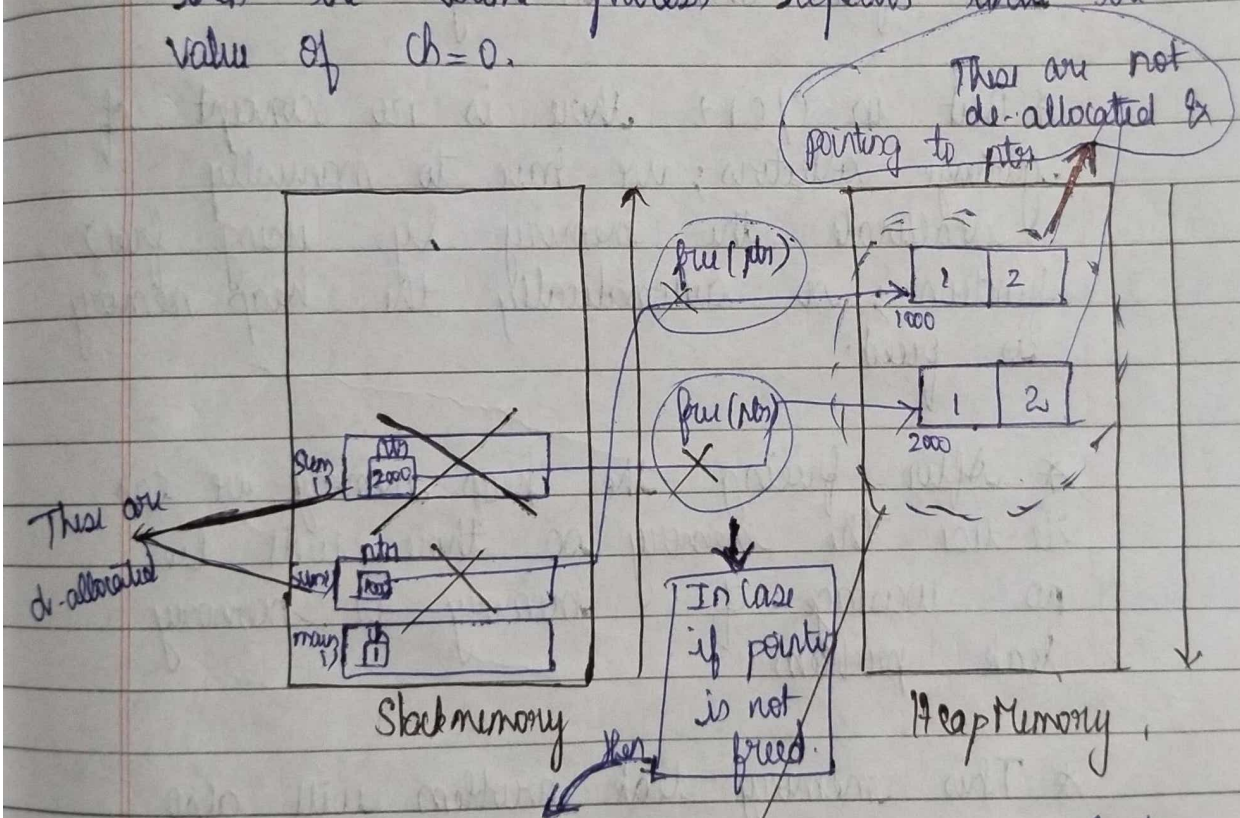
```
int sum()
{
    int *ptr = NULL;
    ptr = (int*) malloc(2 x sizeof(int));
        scanf("%d", ptr+0);
        scanf("%d", ptr+1);
        sum = [*(ptr+0) + *(ptr+1)];
        printf("%d", sum);
}
int main()
{
    int ch=1;
    while (ch)
    {
        sum();
        printf("Continue ?");
        scanf("%d", &ch);
    }
}.
```

* Now the program starts execution from main() and the stack frame for main() is created in stack section. Then inside while loop it calls sum(), now main() is passed and now a separate stack frame for sum() is created.

* But now we allocate a heap section for initializing two variables using pointer inside sum().

* So ptr pointer points to heap section and values are initialized; we do addition & put in 'sum' variable. Now the sum() gets terminated and comes out of its scope, so the allocated memory for sum() stack frame is de-allocated automatically and so the pointer inside it is deallocated; & the heap section allocated memory & initialized contents are still present. but memory is released by pointer so it can be reused

* Now again in main() if user gives ch=1, then the same process repeats until the value of ch=0.



This leads to memory leakage in heap section and we are supposed to deallocate these memory & these memories acts as garbage collectors/values. and it cannot be re-used.

* so memory leakage at some part of time leads to exhaustion of memory.

* But memory is very crucial resource & hence we should not waste the memory.

NOTE:

* So improper use of heap leads to memory leak since we have not properly de-allocated the memory in heap.

NOTE:

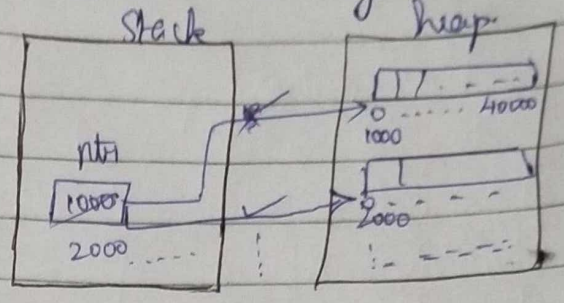
* In Java/C# we have a automatic garbage collector and hence it will identify all those garbage values automatically and it will free those memory.

* But in C/C++ there is no concept of garbage collectors; we have to manually deallocate the memory by using free() function; so automatically the heap memory is freed.

* After freeing the heap memory, we can re-use the memory so there will be no wastage of memory or memory leak problem.

* This memory leak problem will also reduce the performance of the memory and also the memory gets exhausted at some point of time.

Program (3) [This program using DMA; will not free the heap section which leads to memory leak]




```
int main()
{
    int ch=1;
    int *ptr;
    while (ch<50)
    {
        printf("Memory leak demo");
        ptr = (int*) malloc(40000*sizeof(int));
        printf("Continue ?? press 1 for Continue);
        scanf("%d", &ch);
    }                    // free(ptr);
}
```

* Run this program in cmd [gcc filename.c] & [.\a.exe]
  → To see executable file.

* Now open task manager; we can see [a.exe] and the memory consumption is 0.4 MB and now give 1 and enter and till 50.

* Each time the memory consumption is increased; can be seen in task manager.

* Now if you free(ptr); now at every iteration we allocate & free memory; so the memory is not freed which can be seen in task manager.

main.c [11_Memory leak in C] - Code::Blocks 20.03

```c
#include <stdio.h>
#include <stdlib.h>
/** 11-Memory leak in C **/
/** Without using DMA; SMA hence no memory leak **/
void sum()
{
 int a=10,b=11,sum;
 sum=a+b;
 printf("Sum=%d",sum);
}

int main()
{
    int ch=1;

    while(ch)
    {
     sum();
     printf("Continue? Enter 1 to continue...");
     scanf("%d",&ch);
    }
return 0;
}
```

Select Command Prompt - .\a.exe

```
D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMORY ALLOCATION(DMA)\11_Memory leak in C>main.c

D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMORY ALLOCATION(DMA)\11_Memory leak in C>gcc main.c

D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMORY ALLOCATION(DMA)\11_Memory leak in C>.\a.exe
Sum=21Continue? Enter 1 to continue...1
Sum=21Continue? Enter 1 to continue...1
Sum=21Continue? Enter 1 to continue...1
Sum=21Continue? Enter 1 to continue...1
Sum=21Continue? Enter 1 to continue...1
Sum=21Continue? Enter 1 to continue...
```

Command Prompt - .\a.exe

```
D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JE

D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_J

D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_J
Sum=21Continue? Enter 1 to continue...1
Sum=21Continue? Enter 1 to continue...1
Sum=21Continue? Enter 1 to continue...1
Sum=21Continue? Enter 1 to continue...1
Sum=21Continue? Enter 1 to continue...1
Sum=21Continue? Enter 1 to continue...
```

Task Manager

File Options View

Processes | Performance | App history | Startup | Users | Details | Services

Name	Status	3% CPU	62% Memory	15% Disk	0% Network
Apps (6)					
Code::Blocks IDE (2)		0%	15.5 MB	0 MB/s	0 Mbps
crashpad_handler		0%	0.3 MB	0 MB/s	0 Mbps
main.c [11_Memory leak in C] ...		0%	15.2 MB	0 MB/s	0 Mbps
Google Chrome (15)		0.2%	863.2 MB	0.1 MB/s	0 Mbps
Microsoft Edge (10)		0%	214.2 MB	0.1 MB/s	0 Mbps
Task Manager		0.7%	21.4 MB	0 MB/s	0 Mbps
Windows Command Processor (...		0%	11.1 MB	0 MB/s	0 Mbps
a		0%	0.3 MB	0 MB/s	0 Mbps
Command Prompt - .\a.exe		0%	1.7 MB	0 MB/s	0 Mbps
Console Window Host		0%	9.1 MB	0 MB/s	0 Mbps
Windows Explorer (2)		0.3%	41.7 MB	0 MB/s	0 Mbps
Background processes (76)					
64-bit Synaptics Pointing Enhan...		0%	0.1 MB	0 MB/s	0 Mbps
Antimalware Service Executable		0.1%	101.9 MB	0.1 MB/s	0 Mbps

Fewer details | End task

main.c [12_Memory leak in C] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management: Projects | Files | FSymbols

- PART 10_JENNYS LECTURE_DYNA
  - 1_malloc function
  - 2_malloc function for student s
  - 3_calloc function
  - 4_calloc function
  - 5_calloc and malloc function
  - 6_realloc function
  - 7_realloc function
  - 8_realloc function
  - 9_free function
    - Sources
  - 10_free function
    - Sources
      - main.c
  - 11_Memory leak in C
    - Sources
      - main.c
  - **12_Memory leak in C**
    - Sources
      - main.c

```c
#include <stdio.h>
#include <stdlib.h>
/** 12 - Memory leak in C **/
/** Using DMA memory allocated ; but here no memory leak **/
void sum()
{
 int *ptr;
 ptr=(int*) malloc(2*sizeof(int));
 printf("Enter the values to be stored in heap memory allocated");
 scanf("%d",(ptr+0));
 scanf("%d",(ptr+1));
 int sum;
 sum=((*(ptr+0))+(*(ptr+1)));
 printf("Sum=%d",sum);
}
int main()
{
    int ch=1;
    while(ch)
    {
     sum();
     printf("\nContine? Enter 1 to continue... Enter 0 to exit \n");
     scanf("%d",&ch);
    }
    return 0;
}
```

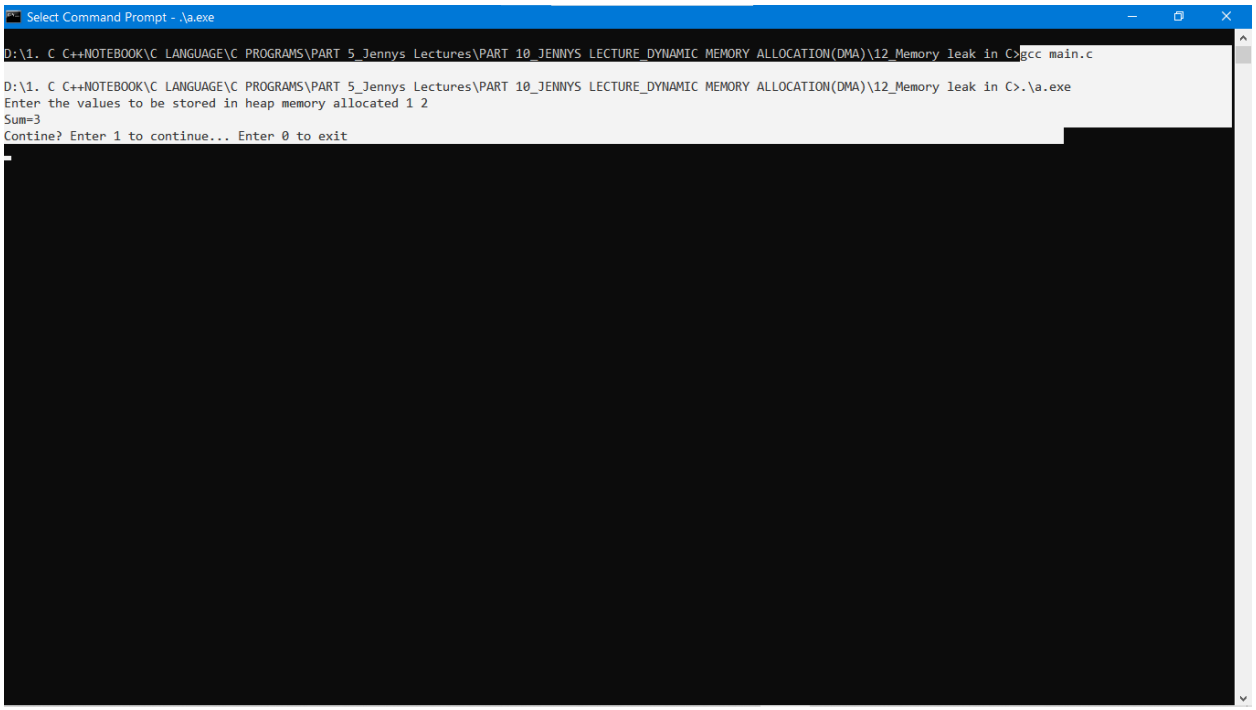
Logs & others: Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++ | CppCheck/Vera++ messages | Cscope

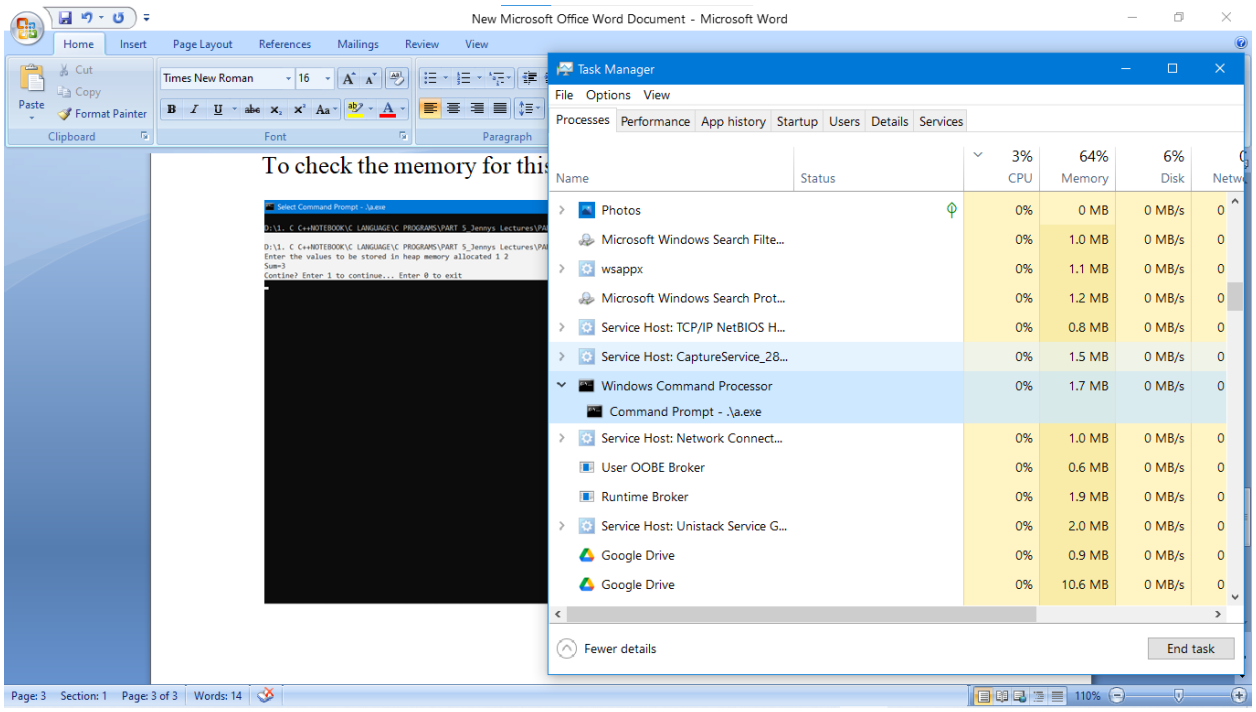
D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRA... | C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 9, Col 67, Pos 263 | Insert | Read/Write | default

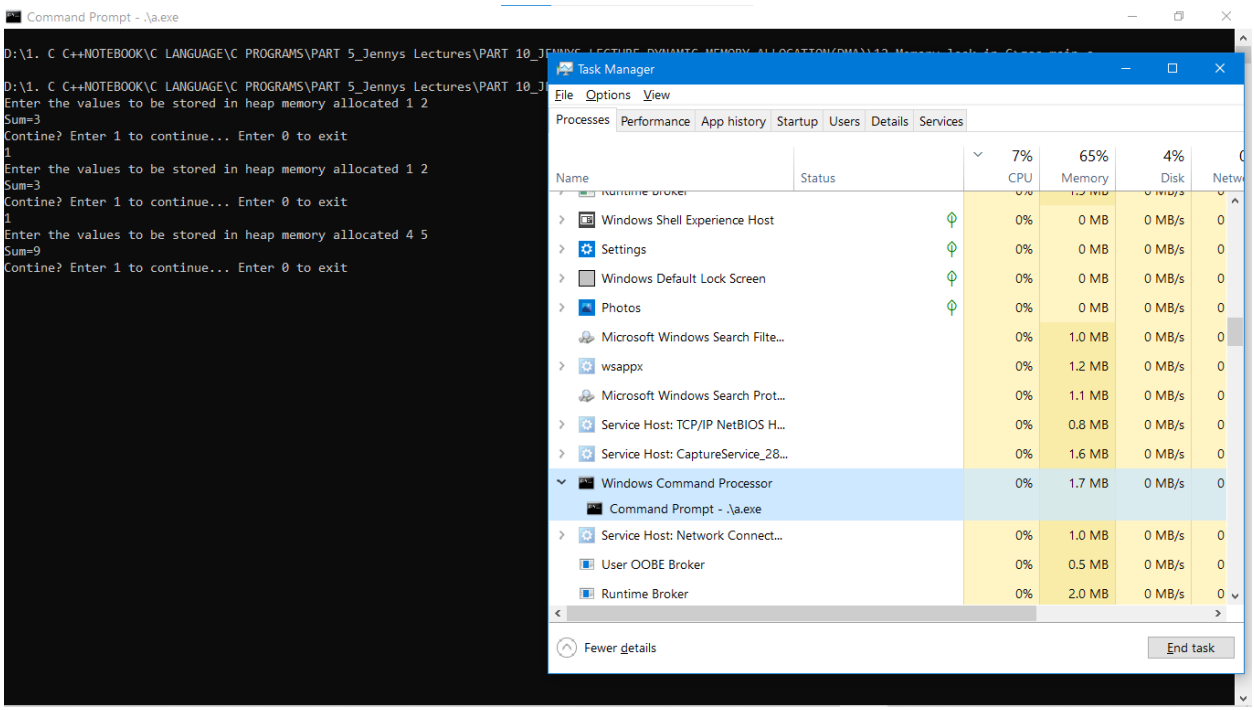
"D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMO

```
Enter the values to be stored in heap memory allocated1 2
Sum=3
Contine? Enter 1 to continue... Enter 0 to exit
```

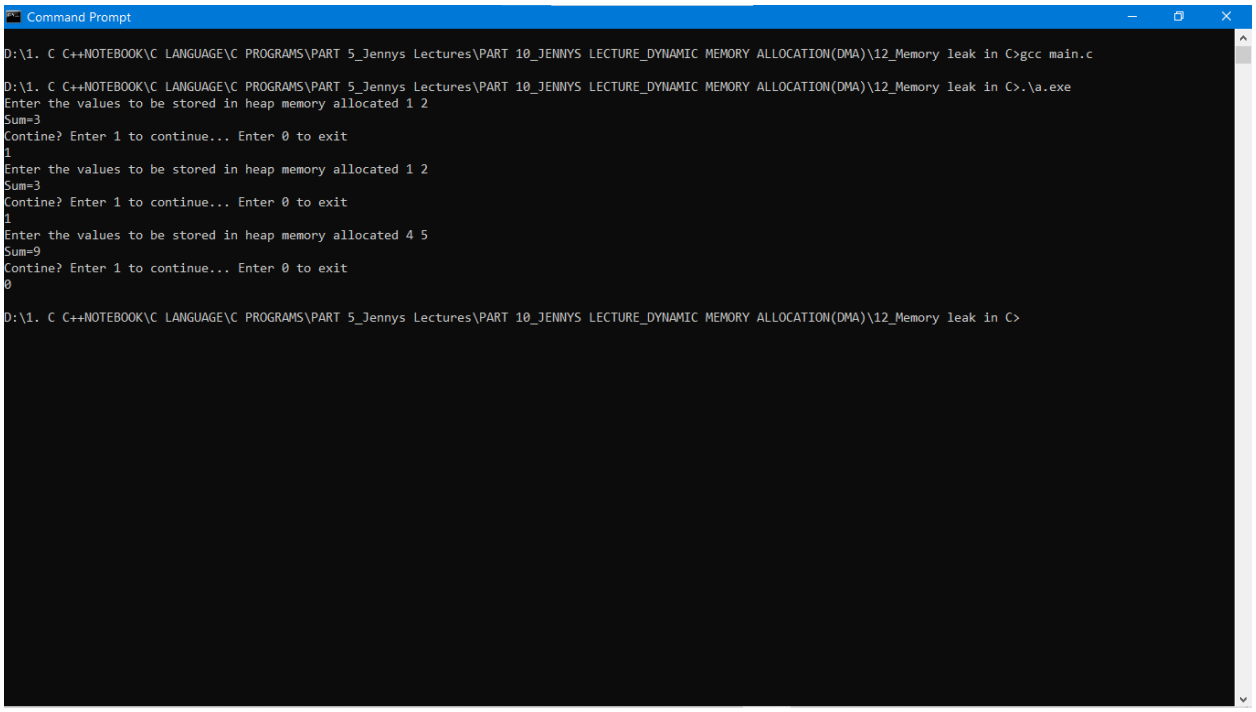
To check the memory for this program run the as executable file in cmd,







Here also no memory leak, since the sum() function memory is de allocated once it comes out of the scope hence the pointer inside it is also de allocated and hence the heap memory is released and every time the new memory is allocated and released.



main.c [13_Memory leak in C] - Code::Blocks 20.03

```c
#include <stdio.h>
#include <stdlib.h>
/** 13 - Memory leak in C **/
int main()
{
    int *ptr,ch=1;
    while(ch)
    {
     printf("\t Memory leak demo \t \n");
     ptr=(int*) malloc(40000*sizeof(int));
     printf("\n Enter 1 to continue, 0 to exit\n");
     scanf("%d",&ch);
    }
return 0;
}
```

Command Prompt - .\a.exe

```
D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMORY ALLOCATION(DMA)\13_Memory leak in C>gcc main.c

D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMORY ALLOCATION(DMA)\13_Memory leak in C>.\a.exe
        Memory leak demo

 Enter 1 to continue, 0 to exit
```

Task Manager

Name	Status	CPU 4%	Memory 64%	Disk 2%	Netw
Service Host: Application Inform...		0%	0.7 MB	0 MB/s	0
Service Host: Radio Manageme...		0%	0.7 MB	0 MB/s	0
Service Host: Network Connecti...		0%	0.6 MB	0 MB/s	0
Google Chrome		0%	0.5 MB	0 MB/s	0
Windows Shell Experience Host		0%	0 MB	0 MB/s	0
Settings		0%	0 MB	0 MB/s	0
Windows Default Lock Screen		0%	0 MB	0 MB/s	0
Windows Command Processor (...		0%	1.1 MB	0 MB/s	0
Command Prompt - .\a.exe		0%	0.8 MB	0 MB/s	0
a		0%	0.3 MB	0 MB/s	0
Service Host: TCP/IP NetBIOS H...		0%	0.3 MB	0 MB/s	0
Service Host: CaptureService_28...		0%	0.8 MB	0 MB/s	0
Service Host: Network Connect...		0%	0.8 MB	0 MB/s	0
User OOBE Broker		0%	0.5 MB	0 MB/s	0
Service Host: Unistack Service G...		0%	1.8 MB	0 MB/s	0

Command Prompt - .\a.exe

```
Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit

1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit

1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit
```

Task Manager

File Options View

Processes | Performance | App history | Startup | Users | Details | Services

Name	Status	4% CPU	64% Memory	6% Disk	Netw
Service Host: Cryptographic Ser...		0%	1.7 MB	0 MB/s	0
COM Surrogate		0%	1.6 MB	0 MB/s	0
Service Host: Application Inform...		0%	0.7 MB	0 MB/s	0
Service Host: Radio Manageme...		0%	0.7 MB	0 MB/s	0
Google Chrome		0%	0.5 MB	0 MB/s	0
Windows Shell Experience Host		0%	0 MB	0 MB/s	0
Settings		0%	0 MB	0 MB/s	0
Windows Default Lock Screen		0%	0 MB	0 MB/s	0
Windows Command Processor (...		0%	1.4 MB	0 MB/s	0
Command Prompt - .\a.exe		0%	0.8 MB	0 MB/s	0
a		0%	0.5 MB	0 MB/s	0
Service Host: TCP/IP NetBIOS H...		0%	0.3 MB	0 MB/s	0
Service Host: Network Connect...		0%	0.8 MB	0 MB/s	0
User OOBE Broker		0%	0.5 MB	0 MB/s	0

Command Prompt - .\a.exe

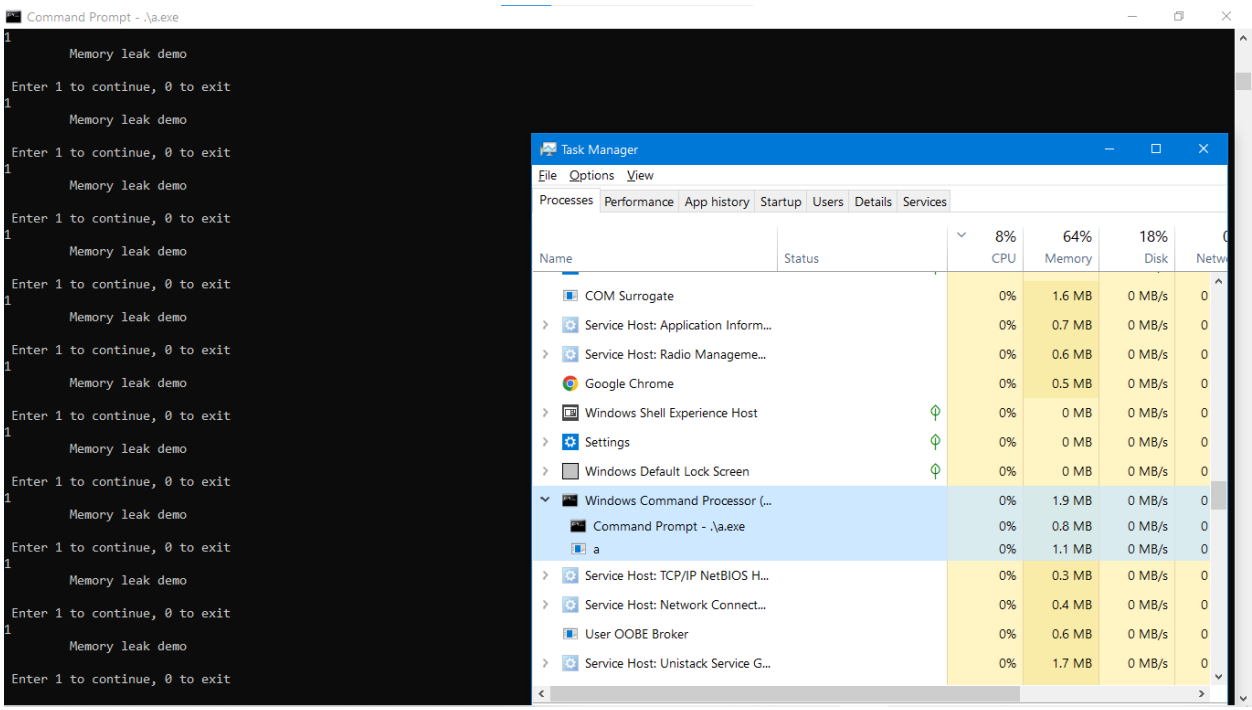
```
1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit
1
        Memory leak demo

Enter 1 to continue, 0 to exit
```

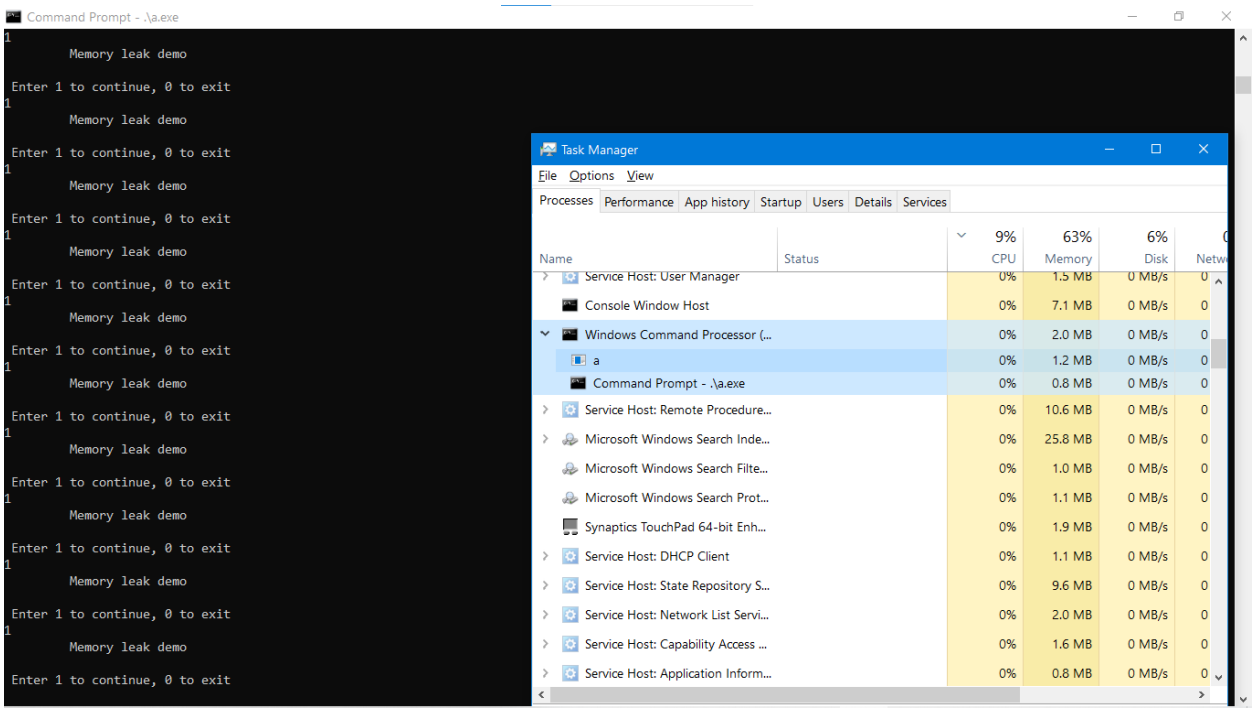
Task Manager

File Options View

Processes | Performance | App history | Startup | Users | Details | Services

Name	Status	13% CPU	64% Memory	2% Disk	Netw
Runtime Broker		0%	1.8 MB	0 MB/s	0
Photos		0%	0 MB	0 MB/s	0
COM Surrogate		0%	1.6 MB	0 MB/s	0
Service Host: Application Inform...		0%	0.7 MB	0 MB/s	0
Service Host: Radio Manageme...		0%	0.6 MB	0 MB/s	0
Google Chrome		0%	0.5 MB	0 MB/s	0
Windows Shell Experience Host		0%	0 MB	0 MB/s	0
Settings		0%	0 MB	0 MB/s	0
Windows Default Lock Screen		0%	0 MB	0 MB/s	0
Windows Command Processor (...		0%	1.7 MB	0 MB/s	0
Command Prompt - .\a.exe		0%	0.8 MB	0 MB/s	0
a		0%	0.9 MB	0 MB/s	0
Service Host: TCP/IP NetBIOS H...		0%	0.3 MB	0 MB/s	0
Service Host: Network Connect...		0%	0.4 MB	0 MB/s	0





So every time we enter the loop new memory is allocated and hence there is memory leak since we did the free the memory every time we enter the loop.

main.c [14_Memory leak in C] - Code::Blocks 20.03

```c
#include <stdio.h>
#include <stdlib.h>
/** 14 - Memory leak in C **/
int main()
{
    int *ptr,ch=1;
    while(ch)
    {
     printf("\t Memory leak demo \t \n");
     ptr=(int*) malloc(40000*sizeof(int));
     printf("\n Enter 1 to continue, 0 to exit\n");
     scanf("%d",&ch);
     free(ptr);
//since we free memory in every iteration; there is no memory leak & memory size is constant
    }

return 0;
}
```

Command Prompt - .\a.exe

```
D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMORY ALLOCATION(DMA)\14_Memory leak in C>gcc main.c

D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10
         Memory leak demo

 Enter 1 to continue, 0 to exit
1
         Memory leak demo

 Enter 1 to continue, 0 to exit
```

Task Manager

Name	Status	CPU 6%	Memory 67%	Disk 10%	Netw
Windows Shell Experience Host		0%	0 MB	0 MB/s	0
Settings		0%	0 MB	0 MB/s	0
Runtime Broker		0%	2.7 MB	0 MB/s	0
Runtime Broker		0%	3.7 MB	0 MB/s	0
Windows Command Processor		0%	0.9 MB	0 MB/s	0
Command Prompt - .\a.exe					
Service Host: Microsoft Account...		0%	3.5 MB	0 MB/s	0
Service Host: TCP/IP NetBIOS H...		0%	0.8 MB	0 MB/s	0
Service Host: Group Policy Client		0%	0.8 MB	0 MB/s	0
Service Host: Windows Time		0%	0.6 MB	0 MB/s	0
User OOBE Broker		0%	0.6 MB	0 MB/s	0
Runtime Broker		0%	2.9 MB	0 MB/s	0
Service Host: Unistack Service G...		0%	1.0 MB	0 MB/s	0
Google Drive		0%	0.9 MB	0 MB/s	0

Fewer details End task

Command Prompt - .\a.exe

```
D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10_JENNYS LECTURE_DYNAMIC MEMORY ALLOCATION(DMA)\14_Memory leak in C>gcc main.c

D:\1. C C++NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lectures\PART 10
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
```

Task Manager

File Options View

Processes | Performance | App history | Startup | Users | Details | Services

Name	Status	3% CPU	67% Memory	6% Disk	Netw
Settings		0%	0 MB	0 MB/s	0
Runtime Broker		0%	2.7 MB	0 MB/s	0
Runtime Broker		0%	3.7 MB	0 MB/s	0
Windows Command Processor		0%	0.9 MB	0 MB/s	0
Command Prompt - .\a.exe					
Service Host: Microsoft Account...		0%	3.4 MB	0 MB/s	0
Service Host: TCP/IP NetBIOS H...		0%	0.8 MB	0 MB/s	0
Service Host: Group Policy Client		0%	0.8 MB	0 MB/s	0
Service Host: Windows Time		0%	0.6 MB	0 MB/s	0
User OOBE Broker		0%	0.5 MB	0 MB/s	0
Service Host: Unistack Service G...		0%	1.0 MB	0 MB/s	0
Google Drive		0%	0.9 MB	0 MB/s	0
Google Drive		0%	12.3 MB	0 MB/s	0
Google Drive		0%	0.1 MB	0 MB/s	0

Fewer details | End task

Command Prompt - .\a.exe

```
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
1
        Memory leak demo

 Enter 1 to continue, 0 to exit
```

Task Manager

File Options View

Processes | Performance | App history | Startup | Users | Details | Services

Name	Status	8% CPU	68% Memory	3% Disk	Netw
Runtime Broker		0%	5.7 MB	0 MB/s	0
Service Host: Remote Desktop S...		0%	1.2 MB	0 MB/s	0
Runtime Broker		0%	2.9 MB	0 MB/s	0
Windows Shell Experience Host		0%	0 MB	0 MB/s	0
Settings		0%	0 MB	0 MB/s	0
Runtime Broker		0%	2.7 MB	0 MB/s	0
Runtime Broker		0%	3.7 MB	0 MB/s	0
Windows Command Processor		0%	0.9 MB	0 MB/s	0
Command Prompt - .\a.exe					
Service Host: Microsoft Account...		0%	3.4 MB	0 MB/s	0
Service Host: TCP/IP NetBIOS H...		0%	0.8 MB	0 MB/s	0
Service Host: Group Policy Client		0%	0.8 MB	0 MB/s	0
Service Host: Windows Time		0%	0.6 MB	0 MB/s	0
User OOBE Broker		0%	0.5 MB	0 MB/s	0

Fewer details | End task