MEMORY
AND
FILE HANDLING

SARTHAK SANAY

(1) AIM:-

To explore the concept of dynamic memory allocation using malloc() and free().

CODE:-

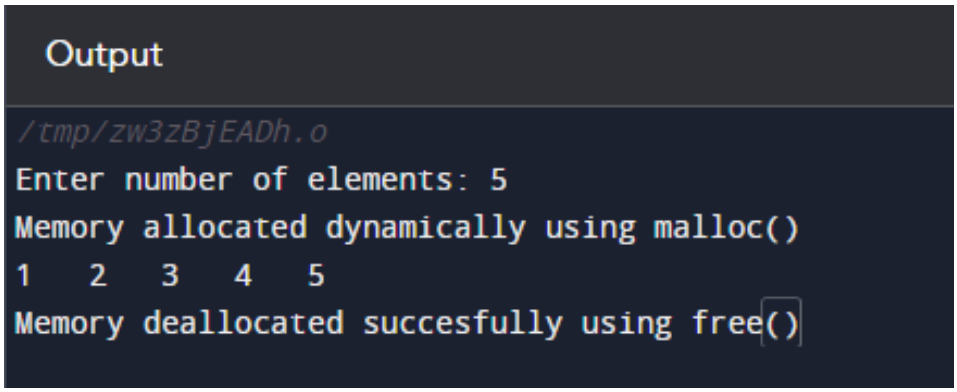
```c
// Program in C to allocate dynamic memory using malloc() and then free the
     memory using free()
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n;
    printf("Enter number of elements: ");
    scanf("%d", &n);

    // Dynamically allocate memory using malloc()
    int *ptr = (int*)malloc(n * sizeof(int));
    if(ptr != NULL)
        printf("Memory allocated dynamically using malloc()\n");
    // Storing the elements of the array
    for(int i=0; i<n; i++)
    {
        ptr[i] = i+1;
    }
    // Printing the elements of the array
    for(int i=0; i<n; i++)
    {
        printf("%d\t", ptr[i]);
    }

    // Deallocating the memory
    free(ptr);
```

```
    printf("Memory deallocated succesfully using free()");
    return 0;
}
```

OUTPUT SCREEN:-



(2) AIM:-

To write programs in C to read from and write to files.

CODE 1:- (Read from files)

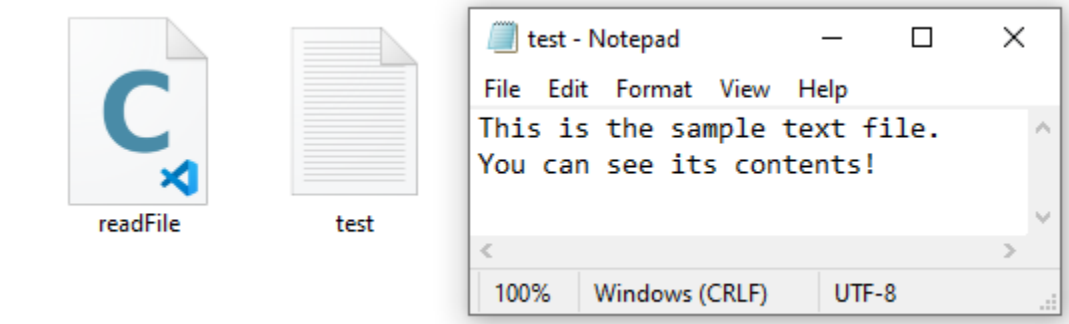
```c
// Program in C to read files
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE* ptr;
    char str[50];
    ptr = fopen("test.txt", "a+");

    if(ptr == NULL)
    {
        printf("Error occured while opening file!\n");
        exit(1);
    }

    printf("Contents of the file are as follows:-\n");
    while(fgets(str, 50, ptr) != NULL)
    {
        printf("%s", str);
    }

    fclose(ptr);
    return 0;
}
```

INPUT FILE:-



OUTPUT SCREEN:-

```
Contents of the file are as follows:-
This is the sample text file.
You can see its contents!
```

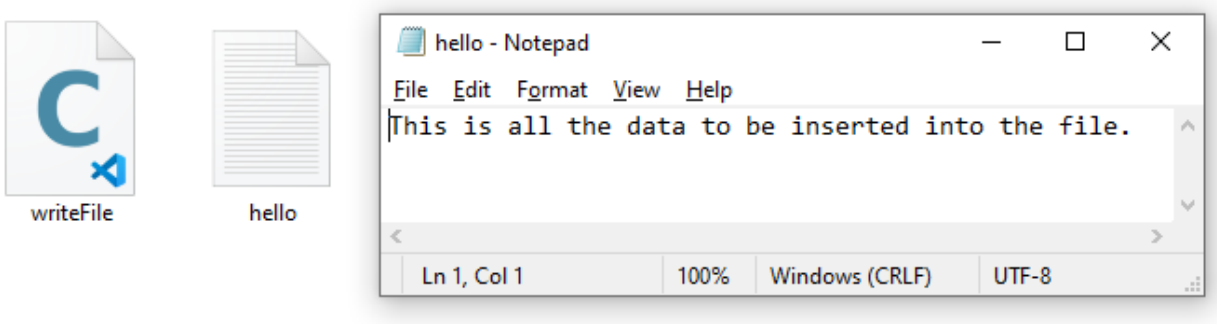
CODE 2:- (Write to files)

```c
// Program in C to write files
#include <stdio.h>
#include <stdlib.h>
int main()
{
    FILE* ptr;
    ptr = fopen("./hello.txt", "w+"); // opens file
    if(ptr == NULL)
    {
        printf("Error occured while writing to file!");
        exit(1);
    }
    char str[] = "This is all the data to be inserted into the file.";
    fputs(str, ptr); // puts data inside the file
    fclose(ptr); // file closed
    printf("Data written inside the file successfully!");
    return 0;
}
```

BEFORE RUNNING PROGRAM:-

writeFile

AFTER RUNNING PROGRAM:-



OUTPUT SCREEN:-

```
Data written inside the file successfully!
```