

Activity 10.2 – Memory Management

Modify the realloc.c by commenting on the realloc function and its corresponding message printing. Then compile and run the realloc.c again.

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
int main(){
    int *ptr;
    ptr = (int *) malloc (5 * sizeof(int));
    if(ptr != NULL){
        printf("Memory has been successfully allocated.\n");
        printf("Starting address: %p\n", ptr);
        printf("End address: %p\n", ptr+4);

        for(int i=0;i<5;i++){
            ptr[i] = i+1;
        }

        //ptr = realloc(ptr, 10 * sizeof(int));
        printf("Successfully reallocated the pointer\n");

        for(int i=5;i<10;i++){
            ptr[i] = i+1;
        }
        printf("The elements of the array are:\n");

        for(int i=0;i<10;i++){
            printf("%d\n",ptr[i]);
        }
    }
}
```

1. What is the result?

```
root@--:~# ./realloc.out
Memory has been successfully allocated.
Starting address: 0x55c4aaca32a0
End address: 0x55c4aaca32b0
Successfully reallocated the pointer
The elements of the array are:
1
2
3
4
5
6
7
8
543492664
909521465
```

Answer: The result is elements inside the array starting from 1 – 8 followed by random number after 8.

2. Why does that happen?

If we commented out the realloc(), we already allocated the 5 previously. But we haven't set the allocation for the rest of it (after 5). Therefore, memory that only reserved only 1-5 and the rest won't. Thus, the unallocated slot might or could be filled with other processes who reserved it first which in my case located after 8 with the value of 5349266 and 90952165.