	COLLEGE OF COMPUTING AND INFORMATION SCIENCES		
	Task # 03		
Class Id	110084	**Course Title**	Operating System
Student Id	**64091**	**Student Name**	**Hassaan Raheem**
Total Marks	05	**Obtained Marks**	

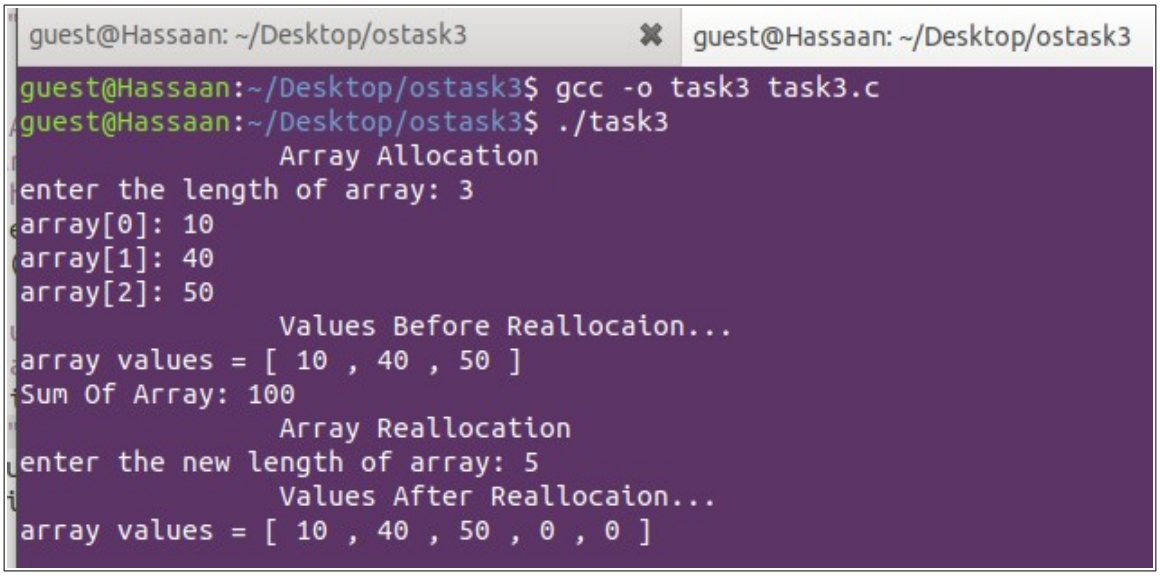
QUESTION:

1 Write a program that allocate memory for array and print the array elements along with sum of all elements. Also reallocate memory size again print array element.

CODE SNIPPET:

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int size , *array;
    printf("\t\tArray Allocation\n");
    printf("enter the length of array: ");
    scanf("%i",&size);
    array = malloc(size * sizeof(int));
    for(int i=0;i<size;i++){
        printf("array[%i]: ",i);
        scanf("%i",&array[i]);
    }

    int sum = 0;

    printf("\t\tValues Before Reallocaion... \n");
    printf("array values = [");
    for(int i=0;i<size;i++){
        printf(" %i ",array[i]);
        sum = sum + array[i];
        if(i==size-1){
            printf("]\n");
        }
        else{
        printf(",");
        }
    }
    printf("Sum Of Array: %i",sum);
    printf("\n\t\tArray Reallocation\n");
    printf("enter the new length of array: ");
    scanf("%i",&size);
    array = realloc(array,size*sizeof(int));

    printf("\t\tValues After Reallocaion... \n");
    printf("array values = [");
    for(int i=0;i<size;i++){
        printf(" %i ",array[i]);
        sum = sum + array[i];
        if(i==size-1){
            printf("]\n");
        }
        else{
        printf(",");
        }
    }



    printf("\n");
    return 0;
}

```

OUTPUT:



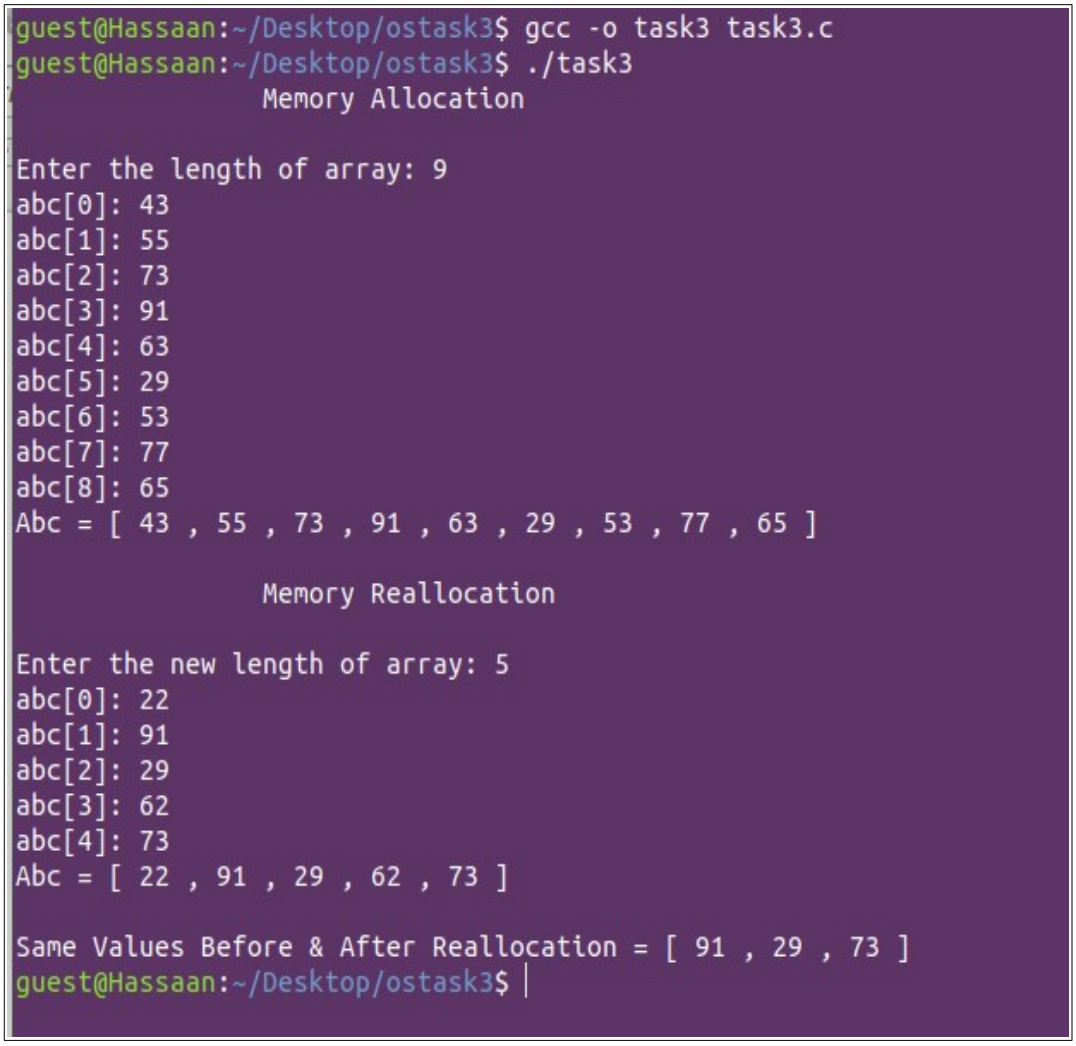
2 Write a C program to dynamically allocate the memory using malloc:

- 2.a User enters the following values in the array.
  - a.i Abc[]={43,55,73,91,63,29,53,77,65}
  - a.ii Print the values after allocating memory.
- 2.b Then reallocate the memory and assign following values.
  - b.i Abc[]={22,91,29,62,73}
  - b.ii Print the values after reallocation.
  - b.iii Find and print the same numbers before and after reallocation.

CODE SNIPPET:

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int length, *abc , *common ;
    printf("\t\tMemory Allocation\n\n");
    printf("Enter the length of array: ");
    scanf("%i",&length);
    int array_record[length];
    abc = malloc(length*sizeof(int));

    //getting input from user
    for(int i=0; i<length; i++)
    {
        printf("abc[%i]: ",i);
        scanf("%i",&abc[i]);
    }

    //printing abc values
    printf("Abc = [");
    for(int i=0; i<length; i++)
    {
        array_record[i] = abc[i];
        if (i==length-1){
            printf(" %d ]",abc[i]);
        }
        else{
            printf(" %d ,",abc[i]);
        }
    }

    printf("\n\n\t\tMemory Reallocation\n\n");
    int new_length;
    printf("Enter the new length of array: ");
    scanf("%i",&new_length);
    abc = realloc(abc,new_length*sizeof(int));

    //getting input from user after reallocation
    for(int i=0; i<new_length; i++)
    {
        printf("abc[%i]: ",i);
        scanf("%i",&abc[i]);
    }

    common = malloc(0*sizeof(int));
    int index = 0;

    //printing abc values after reallocation
    printf("Abc = [");
    for(int i=0; i<new_length; i++)
    {
        for(int j=0; j<length;j++){
            if(array_record[j] == abc[i]){
                common = realloc(common,(index+1)*sizeof(int));
                common[index] = abc[i];
                index ++;
            }
        }

        if (i==new_length-1){
            printf(" %d ]",abc[i]);
        }
        else{
            printf(" %d ,",abc[i]);
        }
    }

    printf("\n\nSame Values Before & After Reallocation = [");
    for(int i=0;i<index;i++){
        if (i==index-1){
            printf(" %d ]",common[i]);
        }
        else{
            printf(" %d ,",common[i]);
        }
    }

    free(abc);
    free(common);


    printf("\n");
    return 0;
}

```

OUTPUT:


guest@Hassaan:~/Desktop/ostask3$ gcc -o task3 task3.c
guest@Hassaan:~/Desktop/ostask3$ ./task3
Memory Allocation
Enter the length of array: 9
abc[0]: 43
abc[1]: 55
abc[2]: 73
abc[3]: 91
abc[4]: 63
abc[5]: 29
abc[6]: 53
abc[7]: 77
abc[8]: 65
Abc = [43 , 55 , 73 , 91 , 63 , 29 , 53 , 77 , 65]
Memory Reallocation
Enter the new length of array: 5
abc[0]: 22
abc[1]: 91
abc[2]: 29
abc[3]: 62
abc[4]: 73
Abc = [22 , 91 , 29 , 62 , 73]
Same Values Before & After Reallocation = [91 , 29 , 73]
guest@Hassaan:~/Desktop/ostask3$