

	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:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int size , *array;
7      printf("\t\tArray Allocation\n");
8      printf("enter the length of array: ");
9      scanf("%i",&size);
10     array = malloc(size * sizeof(int));
11     for(int i=0;i<size;i++){
12         printf("array[%i]: ",i);
13         scanf("%i",&array[i]);
14     }
15
16     int sum = 0;
17
18     printf("\t\tValues Before Reallocaion... \n");
19     printf("array values = [");
20     for(int i=0;i<size;i++){
21         printf(" %i ",array[i]);
22         sum = sum + array[i];
23         if(i==size-1){
24             printf("]\n");
25         }
26         else{
27             printf(",");
28         }
29     }
30     printf("Sum Of Array: %i",sum);
31     printf("\n\t\tArray Reallocation\n");
32     printf("enter the new length of array: ");
33     scanf("%i",&size);
34     array = realloc(array,size*sizeof(int));
35
36     printf("\t\tValues After Reallocaion... \n");
37     printf("array values = [");
38     for(int i=0;i<size;i++){
39         printf(" %i ",array[i]);
40         sum = sum + array[i];
41         if(i==size-1){
42             printf("]\n");
43         }
44         else{
45             printf(",");
46         }
47     }
48
49
50
51
52     printf("\n");
53     return 0;
54 }
55
56

```

OUTPUT:

```
guest@Hassaan: ~/Desktop/ostask3
guest@Hassaan:~/Desktop/ostask3$ gcc -o task3 task3.c
guest@Hassaan:~/Desktop/ostask3$ ./task3
      Array Allocation
Enter the length of array: 3
array[0]: 10
array[1]: 40
array[2]: 50
      Values Before Reallocation...
array values = [ 10 , 40 , 50 ]
Sum Of Array: 100
      Array Reallocation
Enter the new length of array: 5
      Values After Reallocation...
array values = [ 10 , 40 , 50 , 0 , 0 ]
```

- 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:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int length, *abc, *common;
7      printf("\t\tMemory Allocation\n\n");
8      printf("Enter the length of array: ");
9      scanf("%i",&length);
10     int array_record[length];
11     abc = malloc(length*sizeof(int));
12
13     //getting input from user
14     for(int i=0; i<length; i++)
15     {
16         printf("abc[%i]: ",i);
17         scanf("%i",&abc[i]);
18     }
19
20     //printing abc values
21     printf("Abc = [");
22     for(int i=0; i<length; i++)
23     {
24         array_record[i] = abc[i];
25         if (i==length-1){
26             printf(" %d ]",abc[i]);
27         }
28         else{
29             printf(" %d ,",abc[i]);
30         }
31     }
32
33     printf("\n\n\t\tMemory Reallocation\n\n");
34     int new_length;
35     printf("Enter the new length of array: ");
36     scanf("%i",&new_length);
37     abc = realloc(abc,new_length*sizeof(int));
38
39     //getting input from user after reallocation
40     for(int i=0; i<new_length; i++)
41     {
42         printf("abc[%i]: ",i);
43         scanf("%i",&abc[i]);
44     }
45
46     common = malloc(0*sizeof(int));
47     int index = 0;
48
49     //printing abc values after reallocation
50     printf("Abc = [");
51     for(int i=0; i<new_length; i++)
52     {
53         for(int j=0; j<length;j++){
54             if(array_record[j] == abc[i]){
55                 common = realloc(common,(index+1)*sizeof(int));
56                 common[index] = abc[i];
57                 index ++;
58             }
59         }
60
61         if (i==new_length-1){
62             printf(" %d ]",abc[i]);
63         }
64         else{
65             printf(" %d ,",abc[i]);
66         }
67     }
68
69     printf("\n\nSame Values Before & After Reallocation = [");
70     for(int i=0;i<index;i++){
71         if (i==index-1){
72             printf(" %d ]",common[i]);
73         }
74         else{
75             printf(" %d ,",common[i]);
76         }
77     }
78
79     free(abc);
80     free(common);
81
82     printf("\n");
83     return 0;
84 }
85
86
```

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$ |
```