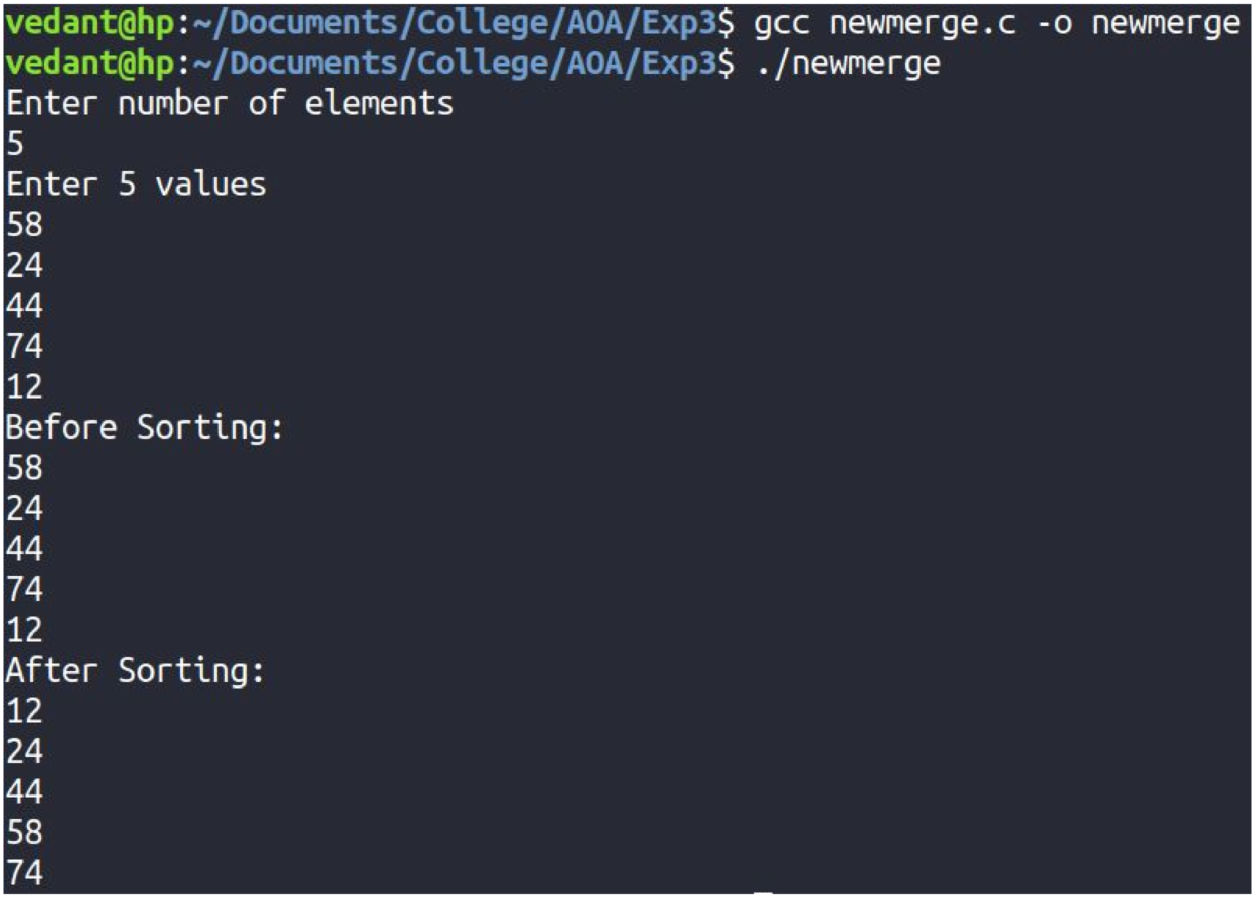
Name:Vedant Mhatre
Moodle ID: 18102055
Roll Number: 31

Modification 1:

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

struct node{
	int data;
	int id;
	struct node *next;
};

struct node *top = NULL;
struct node *ptr = NULL;

void insert()
{
	struct node *new_node = (struct node *) malloc(sizeof(struct node));
	scanf("%d",&new_node->data);
	new_node->next = NULL;

	if (top == NULL){
		new_node->id = 0;
		top = new_node;
		return;
	}
	else{
		struct node *last = top;
		while(last->next != NULL)
			last = last->next;
		new_node->id = (last->id)+1;
		last->next = new_node;
		return;
	}
}

void print()
{
	ptr = top;
	while(ptr!=NULL)
```

```
    {
        printf("%d\n",ptr->data);
        ptr = ptr->next;
    }
}

void ptrAtId(int pid)
{
    ptr = top;
    while(ptr->id!=pid)
        ptr = ptr->next;
}

int ptrData(int pid)
{
    ptrAtId(pid);
    //pid =  ptr->data;
    return ptr->data;
}

void changeData(int pid,int value)
{
    ptrAtId(pid);
    ptr->data = value;
}

void mergearray(struct node *top,int beg,int mid,int end){
    int i,j,k,b[50];

    for(i=beg;i<=end;i++)
        b[i]=ptrData(i);

    i=beg;
    j=mid+1;
    k=beg;

    while((i<=mid)&&(j<=end)) {
        if(b[i]<=b[j])
        {
            changeData(k,b[i]);
            i++;
            k++;
        }
```

```
                else  {
                        changeData(k,b[j]);
                        j++;
                        k++;
                }
        }

        if(i<=mid){
                  while(i<=mid) {
                        changeData(k,b[i]);
                        i++;
                        k++;
                }
        }
        else {
                while(j<=end){
                        changeData(k,b[j]);
                        j++;
                        k++;
                }
        }

}

void mergesort(struct node *top,int beg,int end)
{
int mid;
if(beg<end)
{
        mid=(beg+end)/2;
        mergesort(top,beg,mid);
        mergesort(top,mid+1,end);
        mergearray(top,beg,mid,end);
}
}



void main()
{
        int n;
   printf("Enter number of elements\n");
        scanf("%d",&n);
        printf("Enter %d values\n",n);

        int c;
        for (c = 0; c < n; c++)
                insert();
        printf("Before Sorting:\n");
        print();
        mergesort(top,0,n-1);
        printf("After Sorting:\n");
        print();

}
```



```
vedant@hp:~/Documents/College/AOA/Exp3$ gcc newmerge.c -o newmerge
vedant@hp:~/Documents/College/AOA/Exp3$ ./newmerge
Enter number of elements
5
Enter 5 values
58
24
44
74
12
Before Sorting:
58
24
44
74
12
After Sorting:
12
24
44
58
74
```