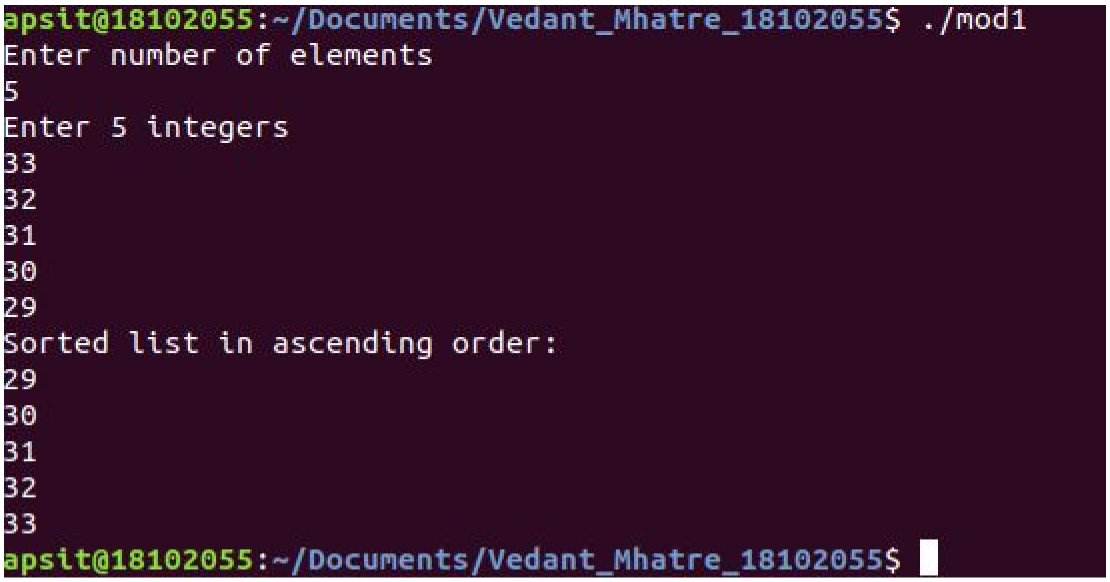
Name : Vedant Mhatre
Moodle Id :18102055
Roll Number : 31

Code:
Modification 1:

```
#include <stdio.h>

void sort(int array[],int n)
{
    int d,c,t;
    for (c = 1 ; c <= n - 1; c++) {
          d = c;

    while ( d > 0 && array[d-1] > array[d]) {
          t= array[d];
          array[d]   = array[d-1];
          array[d-1] = t;

    d--;
          }
    }
}

int main()
{
  int n,c;

  printf("Enter number of elements\n");
  scanf("%d", &n);
```

```
 int array[n];
  printf("Enter %d integers\n", n);

  for (c = 0; c < n; c++)
        scanf("%d", &array[c]);

sort(array,n);

  printf("Sorted list in ascending order:\n");

  for (c = 0; c <= n - 1; c++) {
        printf("%d\n", array[c]);
  }

  return 0;
}
```



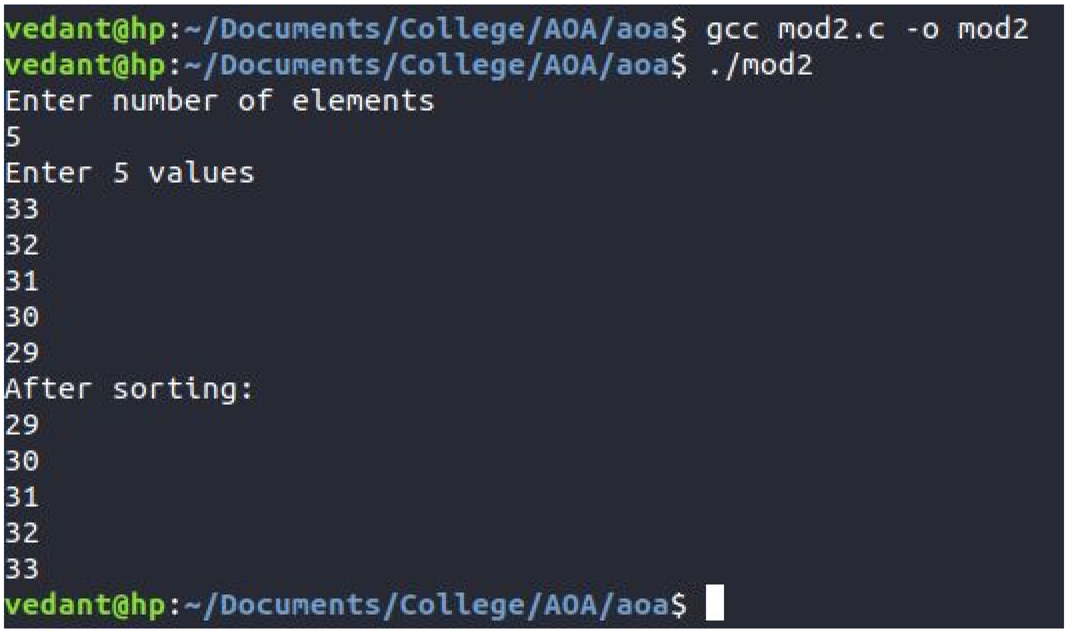
```
apsit@18102055:~/Documents/Vedant_Mhatre_18102055$ ./mod1
Enter number of elements
5
Enter 5 integers
33
32
31
30
29
Sorted list in ascending order:
29
30
31
32
33
apsit@18102055:~/Documents/Vedant_Mhatre_18102055$
```

Modification 2:

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

```
#include <stdlib.h>


void sort(int *array,int *n,int *c)
{
        int *d = (int *)malloc(sizeof(int));
        int *t = (int *)malloc(sizeof(int));
        for(*c=1;*c<=(*n-1);*c=*c+1)
        {       *d = *c;

        while ( *d > 0 && array[*d-1] > array[*d]) {
                *t= array[*d];
                array[*d]   = array[*d-1];
                array[*d-1] = *t;

                *d = *d-1;
                }
        }

}

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

  int *c = (int *)malloc(sizeof(int));
  int array[*n];
```

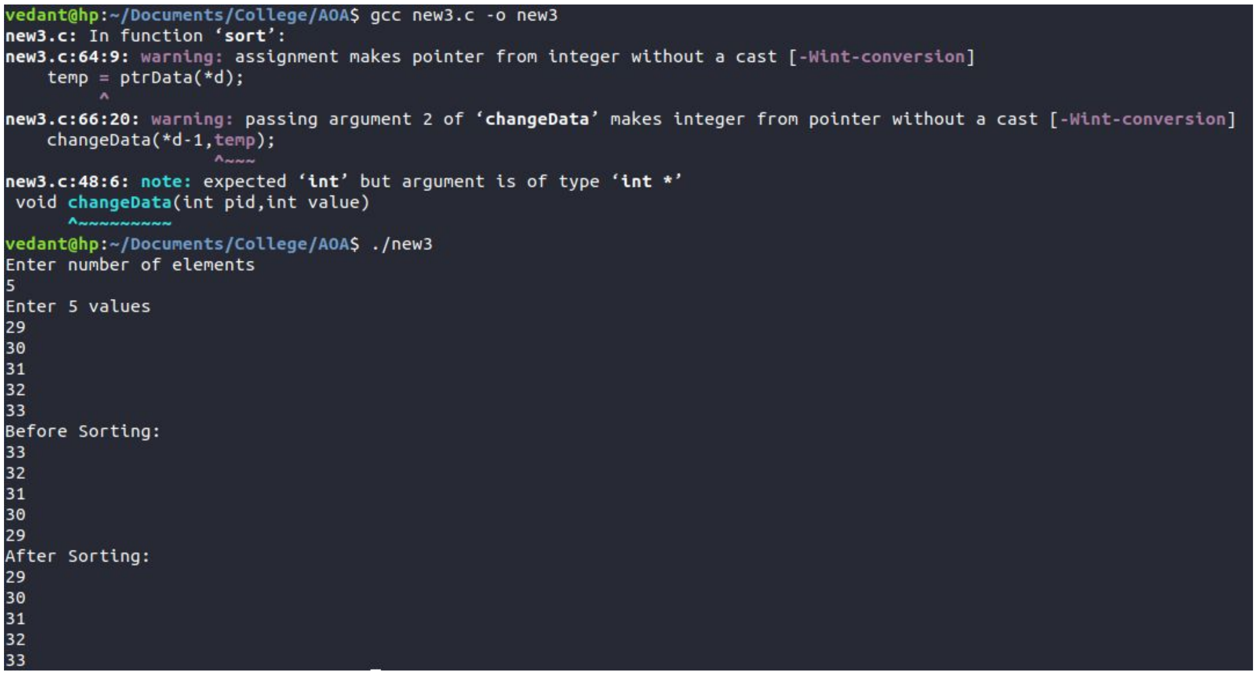
```
        for(*c=0;*c<*n;*c=*c+1)
        scanf("%d",&array[*c]);

        sort(array,n,c);

        printf("After sorting:\n");
  for(*c=0;*c<*n;*c=*c+1)
        printf("%d\n",array[*c]);

}
```



```
vedant@hp:~/Documents/College/AOA/aoa$ gcc mod2.c -o mod2
vedant@hp:~/Documents/College/AOA/aoa$ ./mod2
Enter number of elements
5
Enter 5 values
33
32
31
30
29
After sorting:
29
30
31
32
33
vedant@hp:~/Documents/College/AOA/aoa$ 
```

Modification 3:

```
#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);
        if (top == NULL)
                new_node->id = 0;
        else
                new_node->id = (top->id) + 1;
        new_node->next = top;
        top = new_node;
}

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 pid;
}

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

void sort(struct node *top, int *n, int *c)
{
        int *d = (int *)malloc(sizeof(int));
        int *temp = (int *)malloc(sizeof(int));
        for(*c=*n-1;*c>0;*c=*c-1)
        {
                 *d = *c;

                while(*d<*n && (ptrData(*d)) > (ptrData(*d-1)))
                {
                        temp = ptrData(*d);
                        changeData(*d,ptrData(*d-1));
                        changeData(*d-1,temp);
                        *d = *d+1;
                }

        }
}

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

        int *c = (int *)malloc(sizeof(int));
        for(*c=0;*c<*n;*c=*c+1)
                insert();
        printf("Before Sorting:\n");
        print();
        sort(top,n,c);
        printf("After Sorting:\n");
        print();

}
```



```
vedant@hp:~/Documents/College/AOA$ gcc new3.c -o new3
new3.c: In function ‘sort’:
new3.c:64:9: warning: assignment makes pointer from integer without a cast [-Wint-conversion]
    temp = ptrData(*d);
         ^
new3.c:66:20: warning: passing argument 2 of ‘changeData’ makes integer from pointer without a cast [-Wint-conversion]
    changeData(*d-1,temp);
                    ^~~~
new3.c:48:6: note: expected ‘int’ but argument is of type ‘int *’
 void changeData(int pid,int value)
      ^~~~~~~~~~
vedant@hp:~/Documents/College/AOA$ ./new3
Enter number of elements
5
Enter 5 values
29
30
31
32
33
Before Sorting:
33
32
31
30
29
After Sorting:
29
30
31
32
33
```