

Design and Analysis of Algorithm

1. Write a Program to find the Sum of N Natural numbers using a User defined Function..

```
C n.c
1  /*Write a Program to find the Sum of N natural Numbers using a User defined Function..
2  Prasannaa.V
3  CH.SC.U4CSE24138
4  */
5
6  #include<stdio.h>
7  #include<stdlib.h>
8
9  int natural(int n){
10     int sum=0;
11     for(int i=1;i<=n;i++){
12         sum=sum+i;
13     }
14     return sum;
15 }
16
17
18 int main(){
19     int n;
20     printf("Enter the Number: ");
21     scanf("%d",&n);
22     int x=natural(n);
23     printf("The Sum of the First %d Natural Number is %d ",n,x);
24     printf("\n The Space C0mplexity of this Program is O(1). ");
25
26     return 0;
27 }
28 }
```

```
gcc n.c
prasannaa-vijayakumar@prasannaa-vijayakumar-VirtualBox:~/Documents/DAA$ ./a.out
Enter the Number: 5
The Sum of the First 5 Natural Number is 15
The Space C0mplexity of this Program is O(1). prasannaa-vijayakumar@prasannaa-vija
```

Justification:

The Space Complexity of this program is $O(1)$ because the program uses only a **constant number of variables** (n, x, sum, i).

These variables do **not depend on the input size n**

2. Write a Program to find the sum of Squares of first natural numbers...

```
n1.c
1  /*Write the Program to Find the sum of squares of first Natural numbers
2  Prasannaa.V
3  CH.SC.U4CSE24138
4  */
5
6  #include<stdio.h>
7  #include<stdlib.h>
8
9  int natural(int n){
10     int sum=0;
11     for(int i=1;i<=n;i++){
12         sum=sum+i*i;
13     }
14     return sum;
15 }
16
17
18 int main(){
19     int n;
20     printf("Enter the Number: ");
21     scanf("%d",&n);
22     int x=natural(n);
23     printf("the Sum of Squares of the First %d Natural Number is %d ",n,x);
24     return 0;
25 }
26 }
```

```
s/DAA$ gcc n1.c
prasannaa-vijayakumar@prasannaa-vijayakumar-VirtualBox:~/Documents/DAA$ ./a.out
Enter the Number: 5
The Sum of Squares of the First 5 Natural Number is 55
The Space Complexity of this Program is O(1). prasannaa-vijayakumar@prasannaa-vijayakumar-VirtualBox:~/Documents/DAA
```

Justfication:

The Space Complexity of this program is $O(1)$ because the program uses only a **constant number of variables** (n, x, sum, i).

These variables do **not depend on the input size n**

3. Write the Program to find the sum of cubes of first natural numbers..

```
C n3.c
1  /*Write the Program to Find the sum of Cubes of first Natural numbers
2  Prasannaa.V
3  CH.SC.U4CSE24138
4  */
5  #include<stdio.h>
6  #include<stdlib.h>
7
8  int natural(int n){
9      int sum=0;
10     for(int i=1;i<=n;i++){
11         sum=sum+i*i*i;
12     }
13     return sum;
14 }
15
16
17 int main(){
18     int n;
19     printf("Enter the Number: ");
20     scanf("%d",&n);
21     int x=natural(n);
22     printf("The Sum of Cubes of the First %d Natural Number is %d ",n,x);
23     printf("\n The Space Complexity of this Program is O(1). ");
24
25     return 0;
26 }
27 }
```

```
gcc n3.c
prasannaa-vijayakumar@prasannaa-vijayakumar-VirtualBox:~/Documents/DAA$ ./a.out
Enter the Number: 5
The Sum of Cubes of the First 5 Natural Number is 225
The Space Complexity of this Program is O(1). prasannaa-vijayakumar@prasannaa-vijayakumar-VirtualBox:~/Documents/DAA$
```

Justfication:

The Space Complexity of this program is $O(1)$ because the program uses only a **constant number of variables** (n, x, sum, i).

These variables do **not depend on the input size n**

4. Find the factorial of a Given number??

```
C fact.c
1  /*Find the Factorial of a Given Number
2  Prasannaa.V
3  CH.SC.U4CSE24138
4  */
5  #include<stdio.h>
6  #include<stdlib.h>
7
8  int factorial(int n){
9      if(n==1 || n==0){
10         return 1;
11     }
12     else{
13         return n*factorial(n-1);
14     }
15
16     return n;
17
18 }
19 int main(){
20     int n;
21     printf("Enter the Number :");
22     scanf("%d",&n);
23     int x=factorial(n);
24     printf("%d",x);
25     printf("\n The Space Complexity of the Program is :O(n)");
26
27     return 0;
28 }
```

```
Enter the Number :5
120
The Space Complexity of the Program is :O(n)prasannaa-vijayakumar@prasannaa-vijayakumar-VirtualBox:~/Documents/DAAS$
```

Justification:

The space complexity of this program is **$O(n)$** because the factorial function uses **recursion**, and each recursive call stores its own set of variables (n) in the **function call stack**.

For an input n, the function makes n recursive calls before reaching the base case.

Thus, the call stack grows linearly with n, resulting in **Space Complexity = $O(n)$** .

5. Find the Fibanocci of a Given number??

```
C fib.c
1  /*Find the Fibanocci of a Given Number
2  Prasannaa.V
3  CH.SC.U4CSE24138
4  */
5  #include<stdio.h>
6  #include<stdlib.h>
7
8  int fibanocci(int n){
9      if(n==0){
10         return 0;
11     }
12     else if(n==1){
13         return 1;
14     }
15     else{
16         return fibanocci(n-1)+fibanocci(n-2);
17     }
18
19     return 0;
20 }
21
22
23 int main(){
24     int n;
25     printf("Enter the number: ");
26     scanf("%d",&n);
27     int x=fibanocci(n);
28     printf("%d",x);
29     printf("\n The Space Complexity of this Program is :  $O(n)$ ");
30     return 0;
31 }
```

```
prasannaa-vijayakumar@prasannaa-vijayakumar-VirtualBox:~/Documents/DAA$ gcc fib.c
prasannaa-vijayakumar@prasannaa-vijayakumar-VirtualBox:~/Documents/DAA$ ./a.out
Enter the number: 5
5
The Space Complexity of this Program is : O(n)prasannaa-vijayakumar@prasannaa-vijayakuma
```

Justification:

The Space Complexity of this program is $O(n)$ because the program uses recursion and the maximum number of simultaneously active recursive calls (the recursion depth) is proportional to n ; therefore the stack space grows linearly with n .

6. Write a Program for Transposing a Matrix..

C matrix.c

```
1  /*Write a Program for Transposing the Matrix..
2  Prasannaa.V
3  CH.SC.U4CSE24138
4
5  */
6  |
7  #include<stdio.h>
8  #include<stdlib.h>
9
10 int main(){
11     int rows,columns;
12     printf("Enter the Rows: ");
13     scanf("%d",&rows);
14     printf("Enter the Columns: ");
15     scanf("%d",&columns);
16     int arr[rows][columns];
17
18     for(int i=0;i<rows;i++){
19         for(int j=0;j<columns;j++){
20             printf("Enter the %d %d Elemnt: ",i,j);
21             scanf("%d",&arr[i][j]);
22         }
23     }
24
25     if(rows==columns){
26
27         for(int i=0;i<rows;i++){
28             for(int j=i+1;j<columns;j++){
29                 int temp=arr[i][j];
30                 arr[i][j]=arr[j][i];
31                 arr[j][i]=temp;
32             }
33         }
34     }
```



```

35
36     for(int i=0;i<rows;i++){
37         for(int j=0;j<columns;j++){
38             printf("%d ",arr[i][j]);
39         }
40         printf("\n");
41     }
42
43 }
44 else{
45     printf("Transpose is not Possible..\n");
46 }
47
48     return 0;
49 }

```

```

3 0 0
prasannaa-vijayakumar@prasannaa-vijayakumar-VirtualBox:~/Documents/DAA$ gcc matrix.c
prasannaa-vijayakumar@prasannaa-vijayakumar-VirtualBox:~/Documents/DAA$ ./a.out
Enter the Rows: 3
Enter the Columns: 3
Enter the 0 0 Elemnt: 1
Enter the 0 1 Elemnt: 2
Enter the 0 2 Elemnt: 3
Enter the 1 0 Elemnt: 4
Enter the 1 1 Elemnt: 5
Enter the 1 2 Elemnt: 6
Enter the 2 0 Elemnt: 7
Enter the 2 1 Elemnt: 8
Enter the 2 2 Elemnt: 9
1 4 7
2 5 8
3 6 9

The time Complexity Of the Program is O(Rows*Columns)prasannaa-vijayakumar@prasannaa-vijayakumar-VirtualBox:~/Document
/DAA$

```

Justification:

The space complexity of this program is $O(n^2)$ because it allocates a 2-dimensional array `arr[rows][columns]`, whose size depends on the input.

If the matrix is square ($\text{rows} = \text{columns} = n$), the total memory required is $n \times n$ integers.

Since the memory grows proportionally to n^2 , the space complexity is $O(n^2)$.

Week-2 Sorting Algorithms...

1. Write the Code for the Bubble Sort???

Program:

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  void Display(int arr[],int n){
4      for(int i=0;i<n;i++){
5          printf("%d ",arr[i]);
6      }
7  }
8
9  void bubble(int arr[],int n){
10     for(int i=0;i<n-1;i++){
11         int flag =0;
12         for(int j=0;j<n-1-i;j++){
13             if(arr[j]>arr[j+1]){
14                 int temp=arr[j];
15                 arr[j]=arr[j+1];
16                 arr[j+1]=temp;
17                 flag=1;
18             }
19         }
20     }
21     if(flag==0){
22         break;
23     }
24 }
25
26
27 Display(arr,n);
28
29 }
30
```

```

30
31 int main(){
32     int n;
33     printf("Enter the number of elements :");
34     scanf("%d",&n);
35     int arr[n];
36
37     for(int i=0;i<n;i++){
38         printf("Enter the %d Element: ",i);
39         scanf("%d",&arr[i]);
40
41     }
42
43     bubble(arr,n);
44 }
45

```

Output:

```

15 16 6 8 5 prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ gcc Bubble_Sort.c
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ ./a.out
Enter the number of elements :5
Enter the 0 Element: 15
Enter the 1 Element: 16
Enter the 2 Element: 6
Enter the 3 Element: 8
Enter the 4 Element: 5
5 6 8 15 16 prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$

```

2. Write the Program For the Insertion Sort.

Program:

```
C insertion.c
1  #include<stdio.h>
2  #include<stdio.h>
3
4  void display(int arr[],int n){
5      for(int i=0;i<n;i++){
6          if(i!=n-1){
7              printf("%d,",arr[i]);
8          }
9          else{
10             printf("%d",arr[i]);
11         }
12     }
13 }
14
15 }
16
17
18 void insertion(int arr[],int n){
19     //For the unsorted array:
20     for(int i=1;i<n;i++){
21         int temp=arr[i];
22         int j=i-1;
23
24         //for Sorting inside the sorted array
25         while(j>=0 && arr[j]>temp){
26             arr[j+1]=arr[j];
27             j--;
28         }
29         arr[j+1]=temp;
30     }
31
32     display(arr,n);
33 }
```

```

34 int main(){
35     int n;
36     printf("Enter the number of Elements in the array: ");
37     scanf("%d",&n);
38     int arr[n];
39
40     for(int i=0;i<n;i++){
41         printf("Enter the %d the element: ",i);
42         scanf("%d",&arr[i]);
43     }
44     insertion(arr,n);
45     return 0;
46 }

```

Output:

```

15,16,5,4,3prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ gcc insertion.c
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ ./a.out
Enter the number of Elements in the array: 5
Enter the 0 the element: 15
Enter the 1 the element: 14
Enter the 2 the element: 3
Enter the 3 the element: 2
Enter the 4 the element: 1
1,2,3,14,15prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$

```

3. Write the Program for the Selection Sort ?

Program:

```
C Selection.c
1  #include<stdio.h>
2  #include<stdlib.h>
3  /*Prasannaa.V
4  CH.SC.U4CSE24138
5  Write a Program for Selection Sort??*/
6  |
7
8  void display(int n,int arr[]){
9      for(int i=0;i<n;i++){
10         if(i!=n-1){
11             printf("%d,",arr[i]);
12         }
13         else{
14             printf("%d",arr[i]);
15         }
16     }
17 }
18
19 void Selection(int n,int arr[]){
20     for(int i=0;i<n-1;i++){
21         int min=i;
22
23         for(int j=i+1;j<n;j++){
24             if(arr[j]<arr[min]){
25                 min=j;
26             }
27         }
28         if(min!=i){
29             int temp=arr[i];
30             arr[i]=arr[min];
31             arr[min]=temp;
32         }
33     }
34 }
```

```

34     S
35     |   }
36     }
37
38     display(n,arr);
39 }
40 int main(){
41     int n;
42     printf("Enter the Number of Elements in the Array: ");
43     scanf("%d",&n);
44
45
46     int arr[n];
47     for(int i=0;i<n;i++){
48         printf("Enter %d Element:",i+1);
49         scanf("%d",&arr[i]);
50     }
51
52     Selection(n,arr);
53 }

```

Output:

```

prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ gcc Selection.c
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ ./a.out
Enter the Number of Elements in the Array: 5
Enter 1 Element:8
Enter 2 Element:5
Enter 3 Element:6
Enter 4 Element:7
Enter 5 Element:2
2,5,6,7,8prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$

```

4. Write the code for Bucket Sort???

Program:

```
C Bucket.c
1  /* Prasannaa.V, CH.SC.U4CSE24138 */
2
3  #include <stdio.h>
4  #include <stdlib.h>
5
6  struct Node {
7      float data;
8      struct Node* next;
9  };
10
11 void insert_sorted(struct Node** head, float value) {
12     struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
13     newNode->data = value;
14     newNode->next = NULL;
15
16     if(*head == NULL || (*head)->data >= value) {
17         newNode->next = *head;
18         *head = newNode;
19         return;
20     }
21
22     struct Node* curr = *head;
23     while(curr->next != NULL && curr->next->data < value) {
24         curr = curr->next;
25     }
26
27     newNode->next = curr->next;
28     curr->next = newNode;
29 }
30
```



```

1 void bucket_sort(float arr[], int n) {
2     struct Node* buckets[n];
3
4     for(int i = 0; i < n; i++)
5         buckets[i] = NULL;
6
7     for(int i = 0; i < n; i++) {
8         int idx = arr[i] * n;
9         insert_sorted(&buckets[idx], arr[i]);
10    }
11
12    int k = 0;
13    for(int i = 0; i < n; i++) {
14        struct Node* curr = buckets[i];
15        while(curr != NULL) {
16            arr[k++] = curr->data;
17            curr = curr->next;
18        }
19    }
20 }

```

```

1
2 int main() {
3     int n;
4     printf("Enter number of elements: ");
5     scanf("%d", &n);
6
7     float arr[n];
8     for(int i = 0; i < n; i++) {
9         scanf("%f", &arr[i]);
10    }
11
12    bucket_sort(arr, n);
13
14    printf("Sorted output: ");
15    for(int i = 0; i < n; i++)
16        printf("%g ", arr[i]);
17 }

```

Output:

```
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ gcc Bucket.c
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ ./a.out
Enter number of elements: 5
1
3
7
8
9
Sorted output: 1 3 7 8 9 prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$
```

Program:

```
C Max_heap.c
1  /* Prasannaa.V, CH.SC.U4CSE24138 */
2
3  #include <stdio.h>
4
5  void heapify(int arr[], int n, int i) {
6      int largest = i;
7      int left = 2*i + 1;
8      int right = 2*i + 2;
9
10     if(left < n && arr[left] > arr[largest])
11         largest = left;
12
13     if(right < n && arr[right] > arr[largest])
14         largest = right;
15
16     if(largest != i) {
17         int temp = arr[i];
18         arr[i] = arr[largest];
19         arr[largest] = temp;
20
21         heapify(arr, n, largest);
22     }
23 }
24
25 void heapSort(int arr[], int n) {
26     for(int i = n/2 - 1; i >= 0; i--)
27         heapify(arr, n, i);
28
29     for(int i = n-1; i > 0; i--) {
30         int temp = arr[0];
31         arr[0] = arr[i];
32         arr[i] = temp;
33
34         heapify(arr, i, 0);
35     }
```

```

33
34     heapify(arr, i, 0);
35 }
36 }
37
38 int main() {
39     int n;
40     printf("Enter number of elements: ");
41     scanf("%d", &n);
42
43     int arr[n];
44     for(int i = 0; i < n; i++)
45         scanf("%d", &arr[i]);
46
47     heapSort(arr, n);
48
49     printf("Sorted output: ");
50     for(int i = 0; i < n; i++)
51         printf("%d ", arr[i]);
52 }
53

```

Ln 53, C

Output:

```
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ gcc Max_heap.c
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ ./a.exe
bash: ./a.exe: No such file or directory
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ ./a.out
Enter number of elements: 5
1
5
7
8
2
Sorted output: 1 2 5 7 8 prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$
```

5. Write the code for Min Heap ??

Program:

```

1  /* Prasannaa.V, CH.SC.U4CSE24138 */
2
3  #include <stdio.h>
4  #include<stdlib.h>
5
6
7  #define MAX 100
8
9  int heap[MAX];
10 int heapSize = 0;
11
12 // Swap helper
13 void swap(int *a, int *b) {
14     int temp = *a;
15     *a = *b;
16     *b = temp;
17 }
18
19 // Heapify DOWN (used after deletion)
20 void heapifyDown(int index) {
21     int smallest = index;
22     int left = 2 * index + 1;
23     int right = 2 * index + 2;
24
25     if (left < heapSize && heap[left] < heap[smallest])
26         smallest = left;
27
28     if (right < heapSize && heap[right] < heap[smallest])
29         smallest = right;
30
31     if (smallest != index) {
32         swap(&heap[index], &heap[smallest]);
33         heapifyDown(smallest);
34     }

```

```

38 // Heapify UP (used after insertion)
39 void heapifyUp(int index) {
40     int parent = (index - 1) / 2;
41
42     if (index > 0 && heap[index] < heap[parent]) {
43         swap(&heap[index], &heap[parent]);
44         heapifyUp(parent);
45     }
46 }
47
48 // INSERT into Min Heap
49 void insert(int value) {
50     heap[heapSize] = value;
51     heapifyUp(heapSize);
52     heapSize++;
53 }
54
55 // DELETE MIN (extract root)
56 int extractMin() {
57     if (heapSize <= 0) {
58         printf("Heap is empty!\n");
59         return -1;
60     }
61
62     int minValue = heap[0];
63
64     heap[0] = heap[heapSize - 1];
65     heapSize--;
66
67     heapifyDown(0);
68
69     return minValue;
70 }
71

```

```

72 // DISPLAY HEAP
73 void displayHeap() {
74     printf("Min Heap: ");
75     for (int i = 0; i < heapSize; i++) {
76         printf("%d ", heap[i]);
77     }
78     printf("\n");
79 }
80
81 int main() {
82     int choice, value;
83
84     while (1) {
85         printf("\n---- MIN HEAP MENU ----\n");
86         printf("1. Insert\n");
87         printf("2. Extract Min\n");
88         printf("3. Display Heap\n");
89         printf("4. Exit\n");
90         printf("Choose: ");
91         scanf("%d", &choice);
92
93         switch (choice) {
94             case 1:
95                 printf("Enter value: ");
96                 scanf("%d", &value);
97                 insert(value);
98                 break;
99
100             case 2:
101                 printf("Min removed = %d\n", extractMin());
102                 break;
103
104             case 3:
105                 displayHeap();

```



```

99
100         case 2:
101             printf("Min removed = %d\n", extractMin());
102             break;
103
104         case 3:
105             displayHeap();
106             break;
107
108         case 4:
109             return 0;
110
111         default:
112             printf("Invalid choice!\n");
113     }
114 }
115 }
116

```

Output:

```

prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ gcc Min_Heap.c
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/Sorting W_2$ ./a.out

```

```

---- MIN HEAP MENU ----

```

1. Insert
2. Extract Min
3. Display Heap
4. Exit

Choose: 1

Enter value: 15

```

---- MIN HEAP MENU ----

```

1. Insert
2. Extract Min
3. Display Heap
4. Exit

Choose: 1

Enter value: 95

```

---- MIN HEAP MENU ----

```

1. Insert
2. Extract Min
3. Display Heap

---- MIN HEAP MENU ----

1. Insert
2. Extract Min
3. Display Heap
4. Exit

Choose: 1

Enter value: 89

---- MIN HEAP MENU ----

1. Insert
2. Extract Min
3. Display Heap
4. Exit

Choose: 1

Enter value: 78

---- MIN HEAP MENU ----

1. Insert
2. Extract Min
3. Display Heap
4. Exit

Choose: 1

Enter value: 45

---- MIN HEAP MENU ----

1. Insert
2. Extract Min
3. Display Heap
4. Exit

Choose: 1

Enter value: 25

---- MIN HEAP MENU ----

1. Insert
2. Extract Min
3. Display Heap
4. Exit

Choose: 1

Enter value: 12

---- MIN HEAP MENU ----

1. Insert
2. Extract Min
3. Display Heap
4. Exit

Choose: 1

Enter value: 10

---- MIN HEAP MENU ----

1. Insert
2. Extract Min
3. Display Heap
4. Exit

Choose: 1

Enter value: 1

---- MIN HEAP MENU ----

1. Insert
2. Extract Min
3. Display Heap
4. Exit

Choose: 2

Min removed = 1

```
---- MIN HEAP MENU ----
1. Insert
2. Extract Min
3. Display Heap
4. Exit
Choose: 3
Min Heap: 10 12 15 45 78 89 25 95

---- MIN HEAP MENU ----
1. Insert
2. Extract Min
3. Display Heap
4. Exit
Choose: 4
prasannaa-vijayakumar@Prasannaa:~/Doc
```

Week - 3

1. Write the code for Merge Sort?

Code:

```
W_3 > C merge.c
1  /* Prasannaa.V
2  |   CH.SC.U4CSE24138
3  */
4
5  #include <stdio.h>
6
7  /* Function to merge two sorted subarrays */
8  void merge(int arr[], int low, int mid, int high)
9  {
10     int i = low, j = mid + 1, k = 0;
11     int temp[100];
12
13     while (i <= mid && j <= high)
14     {
15         if (arr[i] <= arr[j])
16             temp[k++] = arr[i++];
17         else
18             temp[k++] = arr[j++];
19     }
20
21     while (i <= mid)
22         temp[k++] = arr[i++];
23
24     while (j <= high)
25         temp[k++] = arr[j++];
26
27     for (i = low, k = 0; i <= high; i++, k++)
28         arr[i] = temp[k];
29 }
30
31 /* Merge Sort function */
32 void mergeSort(int arr[], int low, int high)
33 {
34     if (low < high)
```

```

30
31  /* Merge Sort function */
32  void mergeSort(int arr[], int low, int high)
33  {
34      if (low < high)
35      {
36          int mid = (low + high) / 2;
37
38          mergeSort(arr, low, mid);
39          mergeSort(arr, mid + 1, high);
40          merge(arr, low, mid, high);
41      }
42  }
43
44  int main()
45  {
46      int n, arr[100];
47
48      printf("Enter number of elements: ");
49      scanf("%d", &n);
50
51      printf("Enter elements:\n");
52      for (int i = 0; i < n; i++)
53          scanf("%d", &arr[i]);
54
55      mergeSort(arr, 0, n - 1);
56
57      printf("Sorted array:\n");
58      for (int i = 0; i < n; i++)
59          printf("%d ", arr[i]);
60
61      return 0;
62  }
63

```

Output:

```
Sorted array:
prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/W_3$ ./a.out
Enter number of elements: 12
Enter elements:
157
110
147
122
111
149
151
141
123
112
117
133
Sorted array:
110 111 112 117 122 123 133 141 147 149 151 157 prasannaa-vijayakumar@Prasannaa:~/Documents/DAA/W_3$
```

2. Write the Code for the Quick Sort??

Code:

```
W_3 > C quick.c
1  /* Prasannaa.V
2  | CH.SC.U4CSE24138
3  */
4
5  #include <stdio.h>
6
7  /* Function to swap two elements */
8  void swap(int *a, int *b)
9  {
10     int temp = *a;
11     *a = *b;
12     *b = temp;
13 }
14
15 /* Partition function */
16 int partition(int arr[], int low, int high)
17 {
18     int pivot = arr[low];
19     int i = low + 1;
20     int j = high;
21
22     while (i <= j)
23     {
24         while (i <= high && arr[i] <= pivot)
25             i++;
26
27         while (arr[j] > pivot)
28             j--;
29
30         if (i < j)
31             swap(&arr[i], &arr[j]);
32     }
33
34     swap(&arr[low], &arr[j]);
35     return j;
36 }
```

Ln 68, Col 1


```
W_3 > C quick.c
17 {
18
34     swap(&arr[low], &arr[j]);
35     return j;
36 }
37
38 /* Quick Sort function */
39 void quickSort(int arr[], int low, int high)
40 {
41     if (low < high)
42     {
43         int p = partition(arr, low, high);
44         quickSort(arr, low, p - 1);
45         quickSort(arr, p + 1, high);
46     }
47 }
48
49 int main()
50 {
51     int n, arr[100];
52
53     printf("Enter number of elements: ");
54     scanf("%d", &n);
55
56     printf("Enter elements:\n");
57     for (int i = 0; i < n; i++)
58     {
59         scanf("%d", &arr[i]);
60     }
61
62     quickSort(arr, 0, n - 1);
63
64     printf("Sorted array:\n");
65     for (int i = 0; i < n; i++)
66     {
67         printf("%d ", arr[i]);
68     }
69
70     return 0;
71 }
```

Ln 68, Col 1

Output:

```
110 111 112 117 122 123 133 141 147 149 151 157 prasanna-vijayakumar@Prasannaa:~/Documents/DAA/W_3$ ./a.out
Enter number of elements: 12
Enter elements:
157
110
147
122
111
149
151
141
123
112
117
133
Sorted array:
110 111 112 117 122 123 133 141 147 149 151 157 prasanna-vijayakumar@Prasannaa:~/Documents/DAA/W_3$
```