C++ PROGRAMING

ARRAYS

ARRAY

AN ARRAY IS COLLECTION OF ITEMS STORED AT CONTINUOUS MEMORY LOCATIONS.

ARRAYS ARE USEFUL WHEN WE NEED TO STORE LARGE NUMBER OF INSTANCES WITH THE SAME DATATYPE

“

```
int arr[10]; //specific size
int arr[]={1,2,3}; //specific elements
int arr[3]={1}; //the remaining two elements becomes zero
```

DECLARING ARRAYS

MEMORY ALLOCATION

40	55	63	17	22	68	89	97	89
0	1	2	3	4	5	6	7	8

<- **Array Indices**

Array Length = 9
First Index = 0
Last Index = 8

ARRAY INITIALISING

THIS IS HOW ELEMENTS ARE ADDED IN AN ARRAY

```
int a[10], limit;
cin>>limit;
for(int i=0; i<limit-1; i++)
cin>>a[i];
```

```
int arr[] = {1,4,6,3};
arr[0] //displays '1'
arr[3] //displays '3'
arr[5] //is an error
```

ACCESSING ELEMENTS

TRAVERSING AN ARRAY

GOING THROUGH THE ELEMENTS OF THE GIVEN ARRAY.

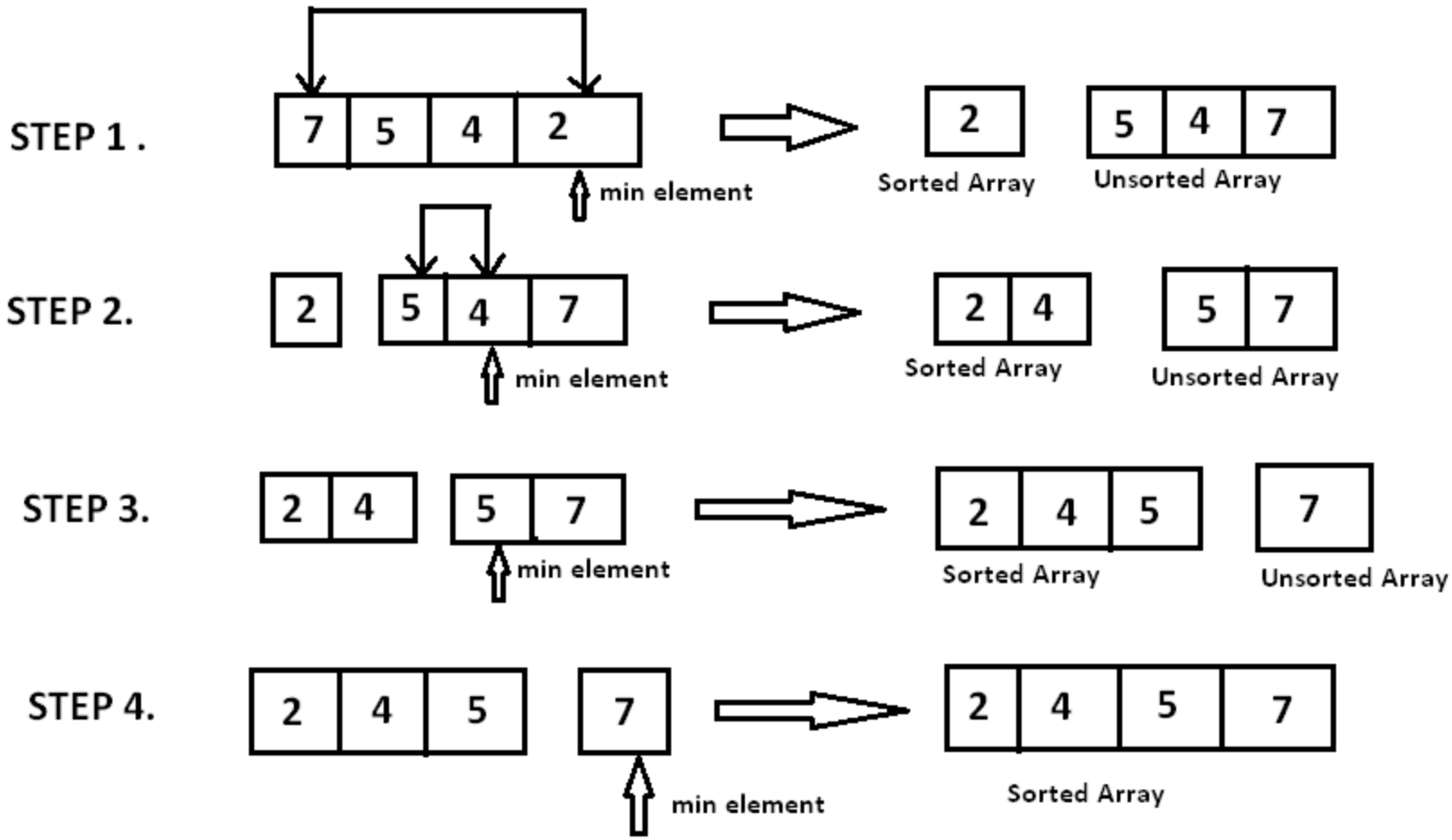
```
int arr[]={1,2,3};
for(int i=0;i<3;i++)
cout<<a[i];
```

BUBBLE SORT

5	1	12	-5	16	unsorted
5	1	12	-5	16	5 > 1, swap
1	5	12	-5	16	5 < 12, ok
1	5	12	-5	16	12 > -5, swap
1	5	-5	12	16	12 < 16, ok
1	5	-5	12	16	1 < 5, ok
1	5	-5	12	16	5 > -5, swap
1	-5	5	12	16	5 < 12, ok
1	-5	5	12	16	1 > -5, swap
-5	1	5	12	16	1 < 5, ok
-5	1	5	12	16	-5 < 1, ok
-5	1	5	12	16	sorted

BUBBLE SORT CODE

```
int temp;
for(int k=0; k<n-1; k++) {
for(int i=0; i<n-k-1; i++) {
if(A[ i ] > A[ i+1] ) {
temp = A[ i ];
A[ i ] = A[ i+1 ];
A[ i + 1] = temp; }
} }
```



SELECTION SORT

SELECTION SORT PROGRAM

```
int min;
for(int i=0; i<n-1; i++) {
min=i;
for(int j=i+1; j<n; j++){
if(A[j] < A[minimum]) {
temp=arr[i];
arr[i]=arr[j];
arr[j]=temp;
} }
```

SEARCHING FOR AN ELEMENT ONE-BY-ONE

```
cout<<"Enter number to be searched";
cin>>num;
for(i=0; i<n; i++) {
if(arr[i]=num)
cout<<"Element found at "<<i+1;
}
```

LINEAR SEARCH

BINARY SEARCH

SEARCH A SORTED ARRAY BY REPEATEDLY DIVIDING THE SEARCH INTERVAL IN HALF.

If searching for 23 in the 10-element array:

	2	5	8	12	16	23	38	56	72	91
23 > 16, take 2nd half	L 2	5	8	12	16	23	38	56	72	H 91
23 < 56, take 1st half	2	5	8	12	16	L 23	38	56	72	H 91
Found 23, Return 5	2	5	8	12	16	L 23	H 38	56	72	91

```
first = 0;
last = n-1;
middle = (first+last)/2;
while (first <= last) {
if(arr[middle] < search)
first = middle + 1;
else if(arr[middle] == search) {
cout<<search<<" found at location "
<<middle+1<<"\n";
break; }
else {last = middle - 1;}
middle = (first + last)/2; }
```

2D ARRAYS

2D ARRAYS ARE ALSO A COLLECTION OF DATA STORED AS A MATRIX (IN ROWS AND COLUMNS)

```
int arr[10][10]; //declaration

//initialising a 2d array
for(i=0; i<n; i++) {
for(j=0; j<n; j++)
cin>>a[i][j]; }
```

THANK YOU