

## Lab 03: Answers

```
#include <iostream>
#include <math.h>

using namespace std;

int* randomArray(int n)
{
    int *A = new int [n];
    for(int i=0; i<n; i++)
    {
        A[i] = rand()%100;
    }
    return A;
}

void displayArray(int *Array, int n)
{
    for(int i=0; i<n; i++)
        cout<<Array[i]<<" ";
    cout<<endl;
}

void swapArray(int* A, int i, int j)
{
    int temp;
    temp = A[i];
    A[i] = A[j];
    A[j] = temp;

    // return A;
}
```

### Implementation of Bubble Sort

```
void bubbleSort(int *A, int n)
{
    for(int i = 0; i<n; i++)
        for(int j = 0; j<n-1; j++)

            if(A[j]>A[j+1])
                swapArray(A, j, j+1);
}
```

```
int *B;

B = randomArray(10);
displayArray(B, 10);

cout<<"*Unsorted Ends* \n";

bubbleSort(B, 10);
displayArray(B, 10);
```

## Implementation of Selection Sort

```
void selectionSort(int * A, int n)
{
    int minVal;        // store minimum
    value
    int i,j,minIndx;

    for(i =0; i<n ; i++)// outer loop
    {
        minVal = A[i];  // initial min val
        minIndx = i;    // initial min ind

        for(j =i; j<n;j++)// inner loop
        {
            if(A[j]<minVal)
            {
                minVal = A[j];
                minIndx = j;
            }
        }

        if(minVal < A[i]) // swap if minimum
            swapArray(A,minIndx,i);
    }
}
```

```
int *B;

B = randomArray(10);
displayArray(B,10);

cout<<"*Unsorted Ends* \n";

selectionSort(B,10);
displayArray(B,10);
```

## Implementation of Insertion Sort

```
void insertionSort(int *A, int n)
{
    int val,j;

    for (int i =1;i<n;i++)
    {
        val = A[i];
        j = i-1;
        while(j>=0 && val<A[j])
        {
            swapArray(A,j,j+1);
            j--;
        }
    }
}
```

```
int *B;

B = randomArray(10);
displayArray(B,10);

cout<<"*Unsorted Ends* \n";

insertionSort(B,10);
displayArray(B,10);
```