ASSIGNMENT 6

Sivasai Sai. Jujjuru
AP19110010547
Cse-H

//CODE 1//

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

#define MAX 7

int intArray[MAX] = {4,6,3,2,1,9,7};

void printline(int count) {
   int i;

   for(i = 0;i < count-1;i++) {
      printf("=");
   }

   printf("=\n");
}

void display() {
   int i;
   printf("[");

   // navigate through all items
   for(i = 0;i < MAX;i++) {
      printf("%d ",intArray[i]);
   }

   printf("]\n");
}

void insertionSort() {

   int valueToInsert;
```

```
   int holePosition;
   int i;

   // loop through all numbers
   for(i = 1; i < MAX; i++) {

      // select a value to be inserted.
      valueToInsert = intArray[i];

      // select the hole position where number is to be inserted
      holePosition = i;

      // check if previous no. is larger than value to be inserted
      while (holePosition > 0 && intArray[holePosition-1] > valueToInsert) {
         intArray[holePosition] = intArray[holePosition-1];
         holePosition--;
         printf(" item moved : %d\n" , intArray[holePosition]);
      }

      if(holePosition != i) {
         printf(" item inserted : %d, at position : %d\n" , valueToInsert,holePosition);
         // insert the number at hole position
         intArray[holePosition] = valueToInsert;
      }

      printf("Iteration %d#:",i);
      display();

   }
}

void main() {
   printf("Input Array: ");
   display();
   printline(50);
   insertionSort();
   printf("Output Array: ");
   display();
   printline(50);
}
```

//CODE 2//

```
#include<stdio.h> // include stdio.h library
#define MAX 5
void bubble_sort(int arr[]); // function declaration

int main(void)
{
   int arr[MAX];

   // input array
   for(int i = 0; i < MAX; i++)
   {
      printf("arr[%d] = ", i);
      scanf("%d", &arr[i]);
   }

   printf("\nUnsorted array: \n");

   // print unsorted array
   for(int i = 0; i < MAX; i++)
   {
      printf("%d ", arr[i]);
   }

   // sort array
   bubble_sort(arr);

   printf("\n\nSorted array: \n");

   // print sorted array
   for(int i = 0; i < MAX; i++)
   {
      printf("%d ", arr[i]);
   }


   return 0; // return 0 to operating system
}

/*
 *  bubble_sort() takes an array and sorts it
 *  in the ascending order
 */
```

```
void bubble_sort(int arr[])
{
   int tmp,  // temporary variable to hold one of the values while swapping
       is_swapped; // variable to indicate whether we have made any swaps during the
passthrough

   for(int i = 0; i < MAX; i++)
   {
      // re-initialize is_swapped to 0 after every passthrough
      is_swapped = 0;

      for(int j = 0; j < MAX - 1 - i; j++)
      {
         if(arr[j] > arr[j+1]) // compare adjacent elements
         {
            // swap adjacent elements
            tmp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = tmp;

            // set is_swapped to 1, to indicate
            // that we have made at least one
            // swap during the passthrough
            is_swapped = 1;
         }
      }

      // if no swaps are made in the last passthrough,
      // exit the outer for loop

      if (!is_swapped)
      {
         break;
      }
   }
}
```

//CODE 3//

```
#include <stdio.h>

void main()
{
```

```
    int A[5],i,j,temp;
    printf("enter 5 elements in the array :");
    for(i=0;i<5,i++)
    {
       scanf("%d",&A[i]);
    }
    printf("\noriginal array:\n");
    for(i=0;i<5;i++)
    printf("%d\t",A[i])   ;

    for(i=0;i<4;i++)
    {
     for(j=i+1;j<5;j++)
     {
        if(A[i]>A[j])
        {
           temp=A[j];
           A[j]=A[i];
           A[i]=temp;
        }
     }
    }
    printf("\nsorted array using selection sort:\n");
    for(i=0;i<5;i++)

    printf("%d\t",A[i])
}
```

//CODE 4//

```
#include <stdio.h>

#define max 10

int a[11] = { 10, 14, 19, 26, 27, 31, 33, 35, 42, 44, 0 };
int b[10];

void merging(int low, int mid, int high) {
   int l1, l2, i;

   for(l1 = low, l2 = mid + 1, i = low; l1 <= mid && l2 <= high; i++) {
      if(a[l1] <= a[l2])
         b[i] = a[l1++];
```

```
      else
        b[i] = a[l2++];
    }

    while(l1 <= mid)
      b[i++] = a[l1++];

    while(l2 <= high)
      b[i++] = a[l2++];

    for(i = low; i <= high; i++)
      a[i] = b[i];
}

void sort(int low, int high) {
    int mid;

    if(low < high) {
      mid = (low + high) / 2;
      sort(low, mid);
      sort(mid+1, high);
      merging(low, mid, high);
    } else {
      return;
    }
}

int main() {
    int i;

    printf("List before sorting\n");

    for(i = 0; i <= max; i++)
      printf("%d ", a[i]);

    sort(0, max);

    printf("\nList after sorting\n");

    for(i = 0; i <= max; i++)
      printf("%d ", a[i]);
}
```