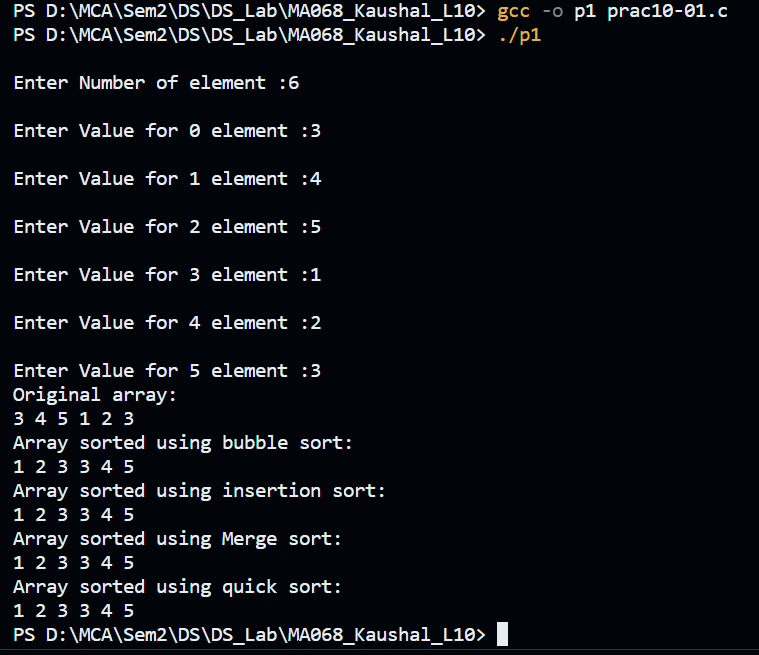
* 1. **Write a Program to collect an unsorted array from the user. Implement sorting of the array using following techniques.** 
     + **bubble sort**
     + **quick sort.**
     + **insertion sort**
     + **Merge sort**

**Code :**

|  |
| --- |
| #include <stdio.h>  void print\_array(int arr[], int n) {  for (int i = 0; i < n; i++) {  printf("%d ", arr[i]);  }  printf("\n");  }  void bubble\_sort(int arr[], int n) {  int temp;  for (int i = 0; i < n-1; i++) {  for (int j = 0; j < n-i-1; j++) {  if (arr[j] > arr[j+1]) {  temp = arr[j];  arr[j] = arr[j+1];  arr[j+1] = temp;  }  }  }  }  void insertion\_sort(int arr[], int n) {  int i, j, key;  for (i = 1; i < n; i++) {  key = arr[i];  j = i - 1;  while (j >= 0 && arr[j] > key) {  arr[j+1] = arr[j];  j--;  }  arr[j+1] = key;  }  }  void merge(int arr[], int left, int mid, int right)  {  int i, j, k;  int s1 = mid - left + 1;  int s2 = right - mid;  int left\_arr[s1], right\_arr[s2];  for (i = 0; i < s1; i++)  left\_arr[i] = arr[left + i];  for (j = 0; j < s2; j++)  right\_arr[j] = arr[mid + 1 + j];    i = 0, j = 0;  k = left;    while (i < s1 && j < s2) {  if (left\_arr[i] <= right\_arr[j]) {  arr[k] = left\_arr[i];  i++;  }  else {  arr[k] = right\_arr[j];  j++;  }  k++;  }    while (i < s1) {  arr[k] = left\_arr[i];  i++;  k++;  }    while (j < s2) {  arr[k] = right\_arr[j];  j++;  k++;  }  }  void merge\_sort(int arr[], int left, int right)  {  if (left < right) {    int mid = left + (right - left) / 2;    merge\_sort(arr, left, mid);    merge\_sort(arr, mid + 1, right);    merge(arr, left, mid, right);  }  }  int partition(int arr[], int low, int high) {  int pivot = arr[high];  int i = low - 1;  int temp;  for (int j = low; j <= high-1; j++) {  if (arr[j] <= pivot) {  i++;  temp = arr[i];  arr[i] = arr[j];  arr[j] = temp;  }  }  temp = arr[i+1];  arr[i+1] = arr[high];  arr[high] = temp;  return i+1;  }  void quick\_sort(int arr[], int low, int high) {  if (low < high) {  int pi = partition(arr, low, high);  quick\_sort(arr, low, pi-1);  quick\_sort(arr, pi+1, high);  }  }  int main() {  int n;    //Create Array  printf("\nEnter Number of element :");  scanf("%d",&n);  int arr[n];  for(int i=0; i<n; i++){  printf("\nEnter Value for %d element :",i);  scanf("%d",&arr[i]);  }  printf("Original array:\n");  print\_array(arr, n);  bubble\_sort(arr, n);  printf("Array sorted using bubble sort:\n");  print\_array(arr, n);  insertion\_sort(arr, n);  printf("Array sorted using insertion sort:\n");  print\_array(arr, n);  merge\_sort(arr,0, n-1);  printf("Array sorted using Merge sort:\n");  print\_array(arr, n);  quick\_sort(arr, 0, n-1);  printf("Array sorted using quick sort:\n");  print\_array(arr, n);  return 0;  } |

**Output :**

****