Rupesh Bhusare, Aditya

Roll no.: 220106, CSE, 2203301, MnC.

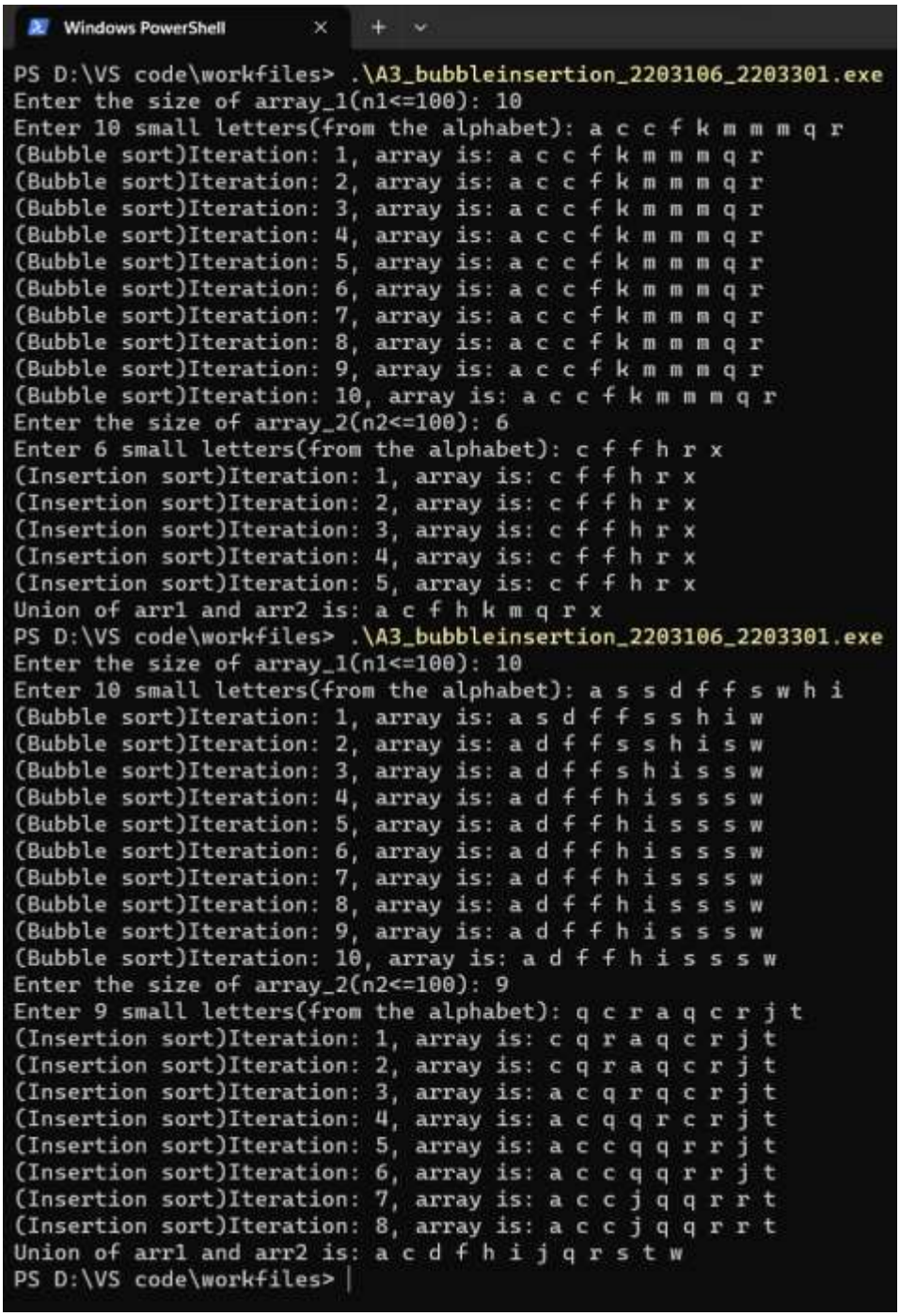
Code:

```c
/*Rupesh Bhusare, Aditya
roll no.: 22030106, CSE, 2203301, MnC*/

#include<stdio.h>
#define MAX_SIZE 100

//printarray function to print elements of array.
void printarray(char arr[], int size_arr){
    for(int i=0; i<size_arr; i++){
        printf("%c ", arr[i]);
    }
    printf("\n");
}

//swap function to interchange the elements btw i and j.
void swap(char arr[], int i, int j){
    char x = arr[i];
    arr[i] = arr[j];
    arr[j] = x;
}

//bubble_sort function is to set all elements in ascending order.
void bubble_sort(char arr[], int size_arr){
    for(int i=0; i<size_arr; i++){
        for(int j=0; j<size_arr-1-i; j++){
            if(arr[j]>arr[j+1]){          //swaping the bigger element at the end of the
array.
                swap(arr, j, j+1);      //swap function is used if j th element is smaller
than i th element.
            }
        }
        printf("(Bubble sort)Iteration: %d, array is: ", i+1);
        printarray(arr, size_arr);    //printing elements of array after every i th
iteration.
    }
}

//insertion_sort function is to set all elements in ascending order.
void insertion_sort(char arr[], int size_arr){
    for(int i=0; i<size_arr-1; i++){
        if(arr[i]>arr[i+1]){
            swap(arr, i, i+1);
            for(int j=i; j>0; j--){
                if(arr[j]<arr[j-1]){        //swapping smaller elements backward.
```

```
                    swap(arr, j, j-1);     //swap function is used if j th element is smaller
than i th element.
                }else{
                    break;
                }
            }
        }
        printf("(Insertion sort)Iteration: %d, array is: ", i+1);
        printarray(arr, size_arr);    //printing elements of array after every i th
iteration.
    }
}
//Union function to take union of arrays of arr1 and arr2 and storing it into arr3.
int Union(char arr1[], char arr2[], char arr3[], int size1, int size2){
    int i=0, j=0, k=0;
    char result[MAX_SIZE];
    while(i<size1 && j<size2){
        if(arr1[i]>arr2[j] ){           //by comparing ith element of arr1 and jth element of
arr2 it adds smaller one into result array.
            result[k++]=arr2[j++];      //so it can maintain result arrays order.
        }else if(arr1[i]<arr2[j]){
            result[k++]=arr1[i++];
        }else{
            result[k++]=arr1[i++];
            j++;
        }
    }
    //while loops to add remaining elements into result array.
    while(i<size1){
        result[k++]=arr1[i++];
    }
    while(j<size2){
        result[k++]=arr2[j++];
    }
    int count=1;
    arr3[0]=result[0];
    for (int i=1;i<k;i++){      //loop to handle duplicate elements,and add all
elements(excluding duplicates) into arr3
        if (result[i]!=result[i-1]){
            arr3[count++]=result[i];
        }
    }

    return count;  //return size of arr3
}

int main(){
    int n1;
    char arr1[MAX_SIZE];
    printf("Enter the size of array_1(n1<=100): ");    //takes inputs of size and elements.
```

```
    scanf("%d", &n1);
    printf("Enter %d small letters(from the alphabet): ", n1);
    for(int i=0; i<n1; i++){
        scanf(" %c",&arr1[i]);
    }
    bubble_sort(arr1, n1);     //sorting with bubble_sort.

    int n2;
    char arr2[MAX_SIZE];
    printf("Enter the size of array_2(n2<=100): ");      //takes inputs of size and elements.
    scanf("%d",&n2);
    printf("Enter %d small letters(from the alphabet): ", n2);
    for(int i=0; i<n2; i++){
        scanf(" %c",&arr2[i]);
    }
    insertion_sort(arr2,n2);   //sorting with insertion_sort.

    char arr3[MAX_SIZE];
    int size_arr3 = Union(arr1, arr2, arr3, n1, n2);    //taking union and returning size of
arr3.
    printf("Union of arr1 and arr2 is: ");
    printarray(arr3, size_arr3);
}
```

Outputs:


Windows PowerShell
PS D:\VS code\workfiles> .\A3_bubbleinsertion_2203106_2203301.exe
Enter the size of array_1(n1<=100): 10
Enter 10 small letters(from the alphabet): a c c f k m m m q r
(Bubble sort)Iteration: 1, array is: a c c f k m m m q r
(Bubble sort)Iteration: 2, array is: a c c f k m m m q r
(Bubble sort)Iteration: 3, array is: a c c f k m m m q r
(Bubble sort)Iteration: 4, array is: a c c f k m m m q r
(Bubble sort)Iteration: 5, array is: a c c f k m m m q r
(Bubble sort)Iteration: 6, array is: a c c f k m m m q r
(Bubble sort)Iteration: 7, array is: a c c f k m m m q r
(Bubble sort)Iteration: 8, array is: a c c f k m m m q r
(Bubble sort)Iteration: 9, array is: a c c f k m m m q r
(Bubble sort)Iteration: 10, array is: a c c f k m m m q r
Enter the size of array_2(n2<=100): 6
Enter 6 small letters(from the alphabet): c f f h r x
(Insertion sort)Iteration: 1, array is: c f f h r x
(Insertion sort)Iteration: 2, array is: c f f h r x
(Insertion sort)Iteration: 3, array is: c f f h r x
(Insertion sort)Iteration: 4, array is: c f f h r x
(Insertion sort)Iteration: 5, array is: c f f h r x
Union of arr1 and arr2 is: a c f h k m q r x
PS D:\VS code\workfiles> .\A3_bubbleinsertion_2203106_2203301.exe
Enter the size of array_1(n1<=100): 10
Enter 10 small letters(from the alphabet): a s s d f f s w h i
(Bubble sort)Iteration: 1, array is: a s d f f s s h i w
(Bubble sort)Iteration: 2, array is: a d f f s s h i s w
(Bubble sort)Iteration: 3, array is: a d f f s h i s s w
(Bubble sort)Iteration: 4, array is: a d f f h i s s s w
(Bubble sort)Iteration: 5, array is: a d f f h i s s s w
(Bubble sort)Iteration: 6, array is: a d f f h i s s s w
(Bubble sort)Iteration: 7, array is: a d f f h i s s s w
(Bubble sort)Iteration: 8, array is: a d f f h i s s s w
(Bubble sort)Iteration: 9, array is: a d f f h i s s s w
(Bubble sort)Iteration: 10, array is: a d f f h i s s s w
Enter the size of array_2(n2<=100): 9
Enter 9 small letters(from the alphabet): q c r a q c r j t
(Insertion sort)Iteration: 1, array is: c q r a q c r j t
(Insertion sort)Iteration: 2, array is: c q r a q c r j t
(Insertion sort)Iteration: 3, array is: a c q r q c r j t
(Insertion sort)Iteration: 4, array is: a c q q r c r j t
(Insertion sort)Iteration: 5, array is: a c c q q r r j t
(Insertion sort)Iteration: 6, array is: a c c q q r r j t
(Insertion sort)Iteration: 7, array is: a c c j q q r r t
(Insertion sort)Iteration: 8, array is: a c c j q q r r t
Union of arr1 and arr2 is: a c d f h i j q r s t w
PS D:\VS code\workfiles>