* Problem **A**

**Bubble Sort**

**AIM :-** To write a ‘c’ Program to take input of Unsorted numbers and display the sorted list of numbers using **Array.**

**Theory :-** The collection of multiple Data in a single variable is known as **Array**. Swapping means exchanging the values between 2 variables. For example Array is initialized as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** Step 1:- Start.

Step 2:- Declare integer variable i, j for loop and ‘a’ for array and ‘b’ as temporary.

Step 3:- Check **FOR** condition i<5, if it is true go to step 4 otherwise go to step 6.

Step 4:- Read variable ‘a’ by **For** loop.

Step 5:- Increment i+1 and go to step 3.

Step 6:- Check **FOR** condition i<=4-1, if it is true go to step 7 otherwise go to step 12.

Step 7:- Check **FOR** condition j<=4-1, if it is true go to step 8 otherwise go to step 11.

Step 8:- Check **if** condition a[j]>a[j+1], if it is true go to step 9 otherwise go to step 10.

Step 9:- Swap the values by b=a[j] then a[j]=a[j+1] and then a[j+1]=b.

Step 10:- Increment j+1 and go to step 7.

Step 11:- Increment i+1 and go to step 6.

Step 12:- Check **FOR** condition i<5, if it is true go to step 13 otherwise go to step 15.

Step 13:- Print variable ‘a’.

Step 14:- Increment i+1 and go to step 12.

Step 15:- Stop.

**Program :-**

#include<stdio.h>

int main()

{

int a[10],b,i,j;

printf("Enter the elements in the Array...\n");

for(i=0;i<5;i++)

scanf("%d",&a[i]);

for(i=0;i<=4-1;i++)

{

for(j=0;j<=4-1;j++)

{

if(a[j]>a[j+1])

{

b=a[j];

a[j]=a[j+1];

a[j+1]=b;

}

}

}

printf("\nAfter Sorting:-\n");

for(i=0;i<5;i++)

printf("%d\t",a[i]);

return 0;

}

**Output :-**

Enter the elements in the Array...

69

45

12

43

76

After Sorting:-

12 43 45 69 76

**Observation :-** After performing the experiment we observed that Values can be stored in a single variable using Array and we were able to take input of some values in an Array using looping statements. The largest value is placed at the End of an Array. Thus, numbers are sorted from Ascending to Descending order. Printing the sorted elements in the same manner using loop. It took 0.16 sec Compilation Time.

* Problem **B**

**Insertion Sort**

**AIM :-** To write a ‘c’ Program to take input of Unsorted numbers and display the sorted list of numbers using **Array.**

**Theory :-** The collection of multiple Data in a single variable is known as **Array**. Swapping means exchanging the values between 2 variables. For example Array is initialized as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** Step 1:- Start.

Step 2:- Declare integer variable i, j for loop and ‘a’ for array and ‘temp’ as temporary.

Step 3:- Check **FOR** condition i<5, if it is true go to step 4 otherwise go to step 6.

Step 4:- Read variable ‘a’ by **For** loop.

Step 5:- Increment i+1 and go to step 3.

Step 6:- Check **FOR** condition i<=5-1, if it is true go to step 7 otherwise go to step 13.

Step 7:- Assign temp=a[i] and j=i-1.

Step 8:- Check **while** condition (j>=0 && a[j]>temp) , if it is true go to step 9 otherwise go to step 11.

Step 9:- Swap the values by a[j+1]=a[j].

Step 10:- Decrement j-1 and go to step 8.

Step 11:- Assign a[j+1]=temp.

Step 12:- Increment i+1 and go to step 6.

Step 13:- Check **FOR** condition i<5, if it is true go to step 14 otherwise go to step 16.

Step 14:- Print variable ‘a’.

Step 15:- Increment i+1 and go to step 12.

Step 16:- Stop.

**Program :-**

#include<stdio.h>

int main()

{

int a[10];

int i,temp,j;

printf("Enter the elements in the Array...\n");

for(i=0;i<5;i++)

scanf("%d",&a[i]);

for(i=1;i<=5-1;i++)

{

temp=a[i];

j=i-1;

while(j>=0 && a[j]>temp)

{

a[j+1]=a[j];

j--;

}

a[j+1]=temp;

}

printf("\nFinal Sort\n");

for(i=0;i<5;i++)

printf("%d\t",a[i]);

return 0;

}

**Output :-**

Enter the elements in the Array...

45

69

23

12

32

Final Sort

12 23 32 45 69

**Observation :-** After performing the experiment we observed that Values can be stored in a single variable using Array and we were able to take input of some values in an Array using looping statements. The Smallest value is placed at the Front of an Array. Thus, numbers are sorted from Ascending to Descending order. Printing the sorted elements in the same manner using loop. It took 0.22 sec Compilation Time.

* Problem **C**

**Selection Sort**

**AIM :-** To write a ‘c’ Program to take input of Unsorted numbers and display the sorted list of numbers using **Array.**

**Theory :-** The collection of multiple Data in a single variable is known as **Array**. Swapping means exchanging the values between 2 variables. For example Array is initialized as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** Step 1:- Start.

Step 2:- Declare variable i, j for loop and ‘a’ for array and ‘temp’ , ‘mini’ & ‘n’.

Step 3:- Read variable ‘n’.

Step 4:- Check **FOR** condition i<n, if it is true go to step 5 otherwise go to step 7.

Step 5:- Read variable ‘a’ by **For** loop.

Step 6:- Increment i+1 and go to step 4.

Step 7:- Check **FOR** condition i<n, if it is true go to step 8 otherwise go to step 15.

Step 8:- Assign mini=i.

Step 9:- Check **FOR** condition j<n , if it is true go to step 10 otherwise go to step 13.

Step 10:- Check **if** condition a[j]<a[mini], if it is true go to step 11 otherwise go to step 12.

Step 11:- Assign mini=j.

Step 12:- Increment j+1 and go to step 9.

Step 13:- Check **if** condition mini!=j, if it is true go to step 14 otherwise go to step 15.

Step 14:- Swap the values by temp=a[j] then a[j]=a[mini] and then a[mini]=temp.

Step 15:- Increment i+1 and go to step 7.

Step 16:- Check **FOR** condition i<n, if it is true go to step 17 otherwise go to step 19.

Step 17:- Print variable ‘a’.

Step 18:- Increment i+1 and go to step 14.

Step 19:- Stop.

**Program :-**

#include<stdio.h>

int main()

{

int a[10];

int i,j,n,mini,temp,k;

printf("How many values you want to Enter???\n");

scanf("%d",&n);

printf("Enter The values in the Array!!\n");

for(i=0;i<n;i++)

scanf("%d",&a[i]);

for(i=0;i<n-1;i++)

{

mini=i;

for(j=i+1;j<n;j++)

{

if(a[j]<a[mini])

{

mini=j;

}

}

if(mini!=i)

{

temp=a[i];

a[i]=a[mini];

a[mini]=temp;

}

}

printf("\n\nSorted Array is.......\n");

for(i=0;i<n;i++)

printf("%d\t",a[i]);

return 0;

}**Output :-**

How many values you want to Enter???

5

Enter The values in the Array!!

54

69

96

87

34

Sorted Array is.......

34 54 69 87 96

**Observation :-** After performing the experiment we observed that Values can be stored in a single variable using Array and we were able to take input of some values in an Array using looping statements. The Smallest value is placed at the Front of an Array. Thus, numbers are sorted from Ascending to Descending order. Printing the sorted elements in the same manner using loop. It took 0.14 sec Compilation Time.

* Problem **D**

**Merge Sort**

**AIM :-** To write a ‘c’ Program to take input of Unsorted numbers and display the sorted list of numbers using **Array** and **Merge sort** Technique.

**Theory :-** The collection of multiple Data in a single variable is known as **Array**. Swapping means exchanging the values between 2 variables. For example Array is initialized as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** UDF(User Defined Function) mergesort ( )

Step 1:- Start.

Step 2:- Receive Values for integer variable ‘A’ for array, ‘low’ and ‘high’ from main function().

Step 3:- Check **if** condition low<high, if it is true go to step 4 otherwise go to step 8.

Step 4:- Declare and Assign integer variable ‘mid’ and mid=(low+high)/2.

Step 5:- Call user defined function “mergesort” **Recursively** by passing ‘A’, ‘low’, ‘mid’ variable’s values as arguments.

Step 6:- Call user defined function “mergesort” **Recursively** by passing ‘A’, ‘mid+1’, ‘high’ variable’s values as arguments.

Step 7:- Call user defined function “merge” by passing ‘A’, ‘low’, ‘mid’, ‘high’ variable’s values as arguments.

Step 8:- Stop.

UDF merge ()

Step 1:- Start.

Step 2:- Receive Values for integer variable ‘A’ for array, ‘low’, ‘mid’ and ‘high’ from mergesort function().

Step 3:- Declare integer variable ‘b’ for array, ‘i’, ‘j’ and ‘k’.

Step 4:- Assign i=low, j=mid+1, k=low.

Step 5:- Check **while** condition **j<=mid && j<=high**, if it is true go to step 6 otherwise go to step 13.

Step 6:- check **if** condition A[i]<=A[j], if it is true go to step 7 otherwise step 9.

Step 7:- Assign b[k]=A[i].

Step 8:- Increment i+1 and go to step 12.

Step 9:- **else,** go to step 10 otherwise step .

Step 10:- Assign b[k]=A[j].

Step 11:- Increment j+1 and go to step 12.

Step 12:- Increment k+1 and go to step 5.

Step 13:- Check **if** condition i>mid, if it is true go to step 14 otherwise go to step 17.

Step 14:- Check **while** condition **j<=high**, if it is true go to step 15 otherwise go to step 17.

Step 15:- Assign b[k]=A[j].

Step 16:- Increment j+1 & k+1 and go to step 14.

Step 17 :- **else** go to step 18 otherwise step 21.

Step 18 :- Check **while** condition **i<=mid**, if it is true go to step 19 otherwise go to step 21.

Step 19 :- Assign b[k]=A[i].

Step 20 :- Increment i+1 & k+1 and go to step 18.

Step 21 :- initialize k=low; Check **For** condition **k<=high**, if it is true go to step 22 otherwise go to step 24.

Step 22:- Assign A[k]=b[k].

Step 23:- Increment k+1 and go to step 21.

Step 24:- Stop.

UDF printarray()

Step 1:- Start.

Step 2:- Receive Values for integer variable ‘A’ for array, ‘size’ from main function().

Step 3:- Declare ‘i’ variable for loop.

Step 4:- initialize i=0; Check **For** condition **i<size**, if it is true go to step 5 otherwise go to step 6.

Step 5:- Print values of variable ‘A’.

Step 6:- Stop.

Main function ()

Step 1:- Start.

Step 2:- Declare integer variable ‘i’ and ‘n’.

Step 3:- Read value for ‘n’ variable.

Step 4:- Declare ‘a’ for Array.

Step 5:- initialize i=0; Check **For** condition **i<n**, if it is true go to step 6 otherwise go to step 7.

Step 6:- Read value for ‘a’ by loop.

Step 7:- Call the printArray function and pass the arguments ‘a’ and ‘n’.

Step 8:- Call the mergesort function and pass arguments ‘a’, ‘0’ and ‘n-1’.

Step 9:- Call again the printArray function and the arguments ‘a’ and ‘n’.

Step 10:- Stop.

**Program :-**

#include <stdio.h> //Merge sort

void mergeSort(int A[], int low, int high)

{

if(low<high)

{

int mid=(low+high)/2;

mergeSort(A,low,mid);

mergeSort(A,mid+1,high);

merge(A,low,mid,high);

}

}

void merge(int A[], int low, int mid, int high)

{

int b[high];

int i, j, k;

i=low;

j=mid+1;

k=low;

while(i<=mid && j<=high)

{

if(A[i]<=A[j])

{

b[k]=A[i];

i++;

}

else

{

b[k]=A[j];

j++;

}

k++;

}

if(i>mid)

{

while(j<=high)

{

b[k]=A[j];

j++;

k++;

}

}

else

{

while(i<=mid)

{

b[k]=A[i];

i++;

k++;

}

}

for(k=low;k<=high;k++)

A[k]=b[k];

}

void printArray(int A[], int size)

{

int i;

for(i=0;i<size;i++)

printf("%d ",A[i]);

printf("\n");

}

int main()

{

int i,n;

printf("How Many values you want to Enter???\n");

scanf("%d",&n);

int a[n];

printf("Enter the Values in the Array???\n");

for(i=0;i<n;i++)

scanf("%d",&a[i]);

printf("\n\nGiven array is \n");

printArray(a, n);

mergeSort(a, 0, n-1);

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

printArray(a, n);

return 0;

}

**Output :-**

How Many values you want to Enter???

5

Enter the Values in the Array???

9

8

7

6

5

Given array is

9 8 7 6 5

Sorted array is

5 6 7 8 9

**Observation :-** After performing the experiment we observed that Values can be stored in a single variable using Array and we were able to take input of some values in an Array using looping statements. The Smallest value is placed at the Front of an Array. Thus, numbers are sorted from Ascending to Descending order. Printing the sorted elements in the same manner using loop. It took 0.31 sec Compilation Time.