1. *\* C program to accept N numbers and arrange them in an ascending order  \*/*
3. #include <stdio.h>
4. void main()
5. {
6. int i, j, a, n, number[30];
7. printf("Enter the value of N **\n**");
8. scanf("%d", &n);
9. printf("Enter the numbers **\n**");
10. for (i = 0; i < n; ++i)
11. scanf("%d", &number[i]);
13. for (i = 0; i < n; ++i)
14. {
15. for (j = i + 1; j < n; ++j)
16. {
17. if (number[i] > number[j])
18. {
19. a = number[i];
20. number[i] = number[j];
21. number[j] = a;
22. }
23. }
24. }
26. printf("The numbers arranged in ascending order are given below **\n**");
27. for (i = 0; i < n; ++i)
28. printf("%d**\n**", number[i]);

}

1. #include <stdio.h>
2. #include <conio.h>
3. void main()
4. {
5. int a[2][3],b[2][3],c[2][3],i,j;
6. clrscr();
7. printf("\nENTER VALUES FOR MATRIX A:\n");
8. for(i=0;i<2;i++)
9. for(j=0;j<3;j++)
10. scanf("%d",&a[i][j]);
11. printf("\nENTER VALUES FOR MATRIX B:\n");
12. for(i=0;i<2;i++)
13. for(j=0;j<3;j++)
14. scanf("%d",&b[i][j]);
15. for(i=0;i<2;i++)
16. for(j=0;j<3;j++)
17. c[i][j]=a[i][j]+b[i][j];
18. printf("\nTHE VALUES OF MATRIX C ARE:\n");
19. for(i=0;i<2;i++)
20. {
21. for(j=0;j<3;j++)
22. printf("%5d",c[i][j]);
23. printf("\n");
24. }
25. getch();
26. }
27. *C Program to Merge the Elements of 2 Sorted Array   \*/*
28. #include <stdio.h>
29. void main()
30. {
31. int array1[50], array2[50], array3[100], m, n, i, j, k = 0;
32. printf("**\n** Enter size of array Array 1: ");
33. scanf("%d", &m);
34. printf("**\n** Enter sorted elements of array 1: **\n**");
35. for (i = 0; i < m; i++)
36. {
37. scanf("%d", &array1[i]);
38. }
39. printf("**\n** Enter size of array 2: ");
40. scanf("%d", &n);
41. printf("**\n** Enter sorted elements of array 2: **\n**");
42. for (i = 0; i < n; i++)
43. { scanf("%d", &array2[i]); }
44. i = 0;
45. j = 0;
46. while (i < m && j < n)
47. {
48. if (array1[i] < array2[j])
49. {
50. array3[k] = array1[i];
51. i++;
52. }
53. else
54. {
55. array3[k] = array2[j];
56. j++;
57. }
58. k++;
59. }
60. if (i >= m)
61. {
62. while (j < n)
63. {
64. array3[k] = array2[j];
65. j++;
66. k++;
67. }
68. }
69. if (j >= n)
70. {
71. while (i < m)
72. {
73. array3[k] = array1[i];
74. i++;
75. k++;
76. }
77. }
78. printf("**\n** After merging: **\n**");
79. for (i = 0; i < m + n; i++)
80. {
81. printf("**\n**%d", array3[i]);
82. } }

**Program Explanation**

1. Declare 2 1D arrays of some fixed size, then take size of the arrays from user and define all the elements of the array according to the size in sorted fashion.  
   2. Take two variables, i and j as iterators which will track the position of elements in arrays.  
   3. Running a while loop till we reach the end of either array, the element at ith and jth position of two arrays are compared.  
   4. The smaller element gets inserted into final array (third array, whose size is the sum of the size of these two arrays) and the track position gets incremented by 1.  
   5. This process continues, till we reach the end of either array.  
   6. After finishing the loop above, one of the array’s tracker(i.e either i or j) will not be at the last position of the corresponding array, in that case we will have to add all the remaining elements of that array to the final array as it is one by one

**Runtime Test Cases**

1. Enter size of array Array 1: 4
2. Enter sorted elements of array 1:
3. 12
4. 18
5. 40
6. 60
7. Enter size of array 2: 4
8. Enter sorted elements of array 2:
9. 47
10. 56
11. 89
12. 90
13. After merging:
14. 12
15. 18
16. 40
17. 47
18. 56
19. 60
20. 89
21. 90