

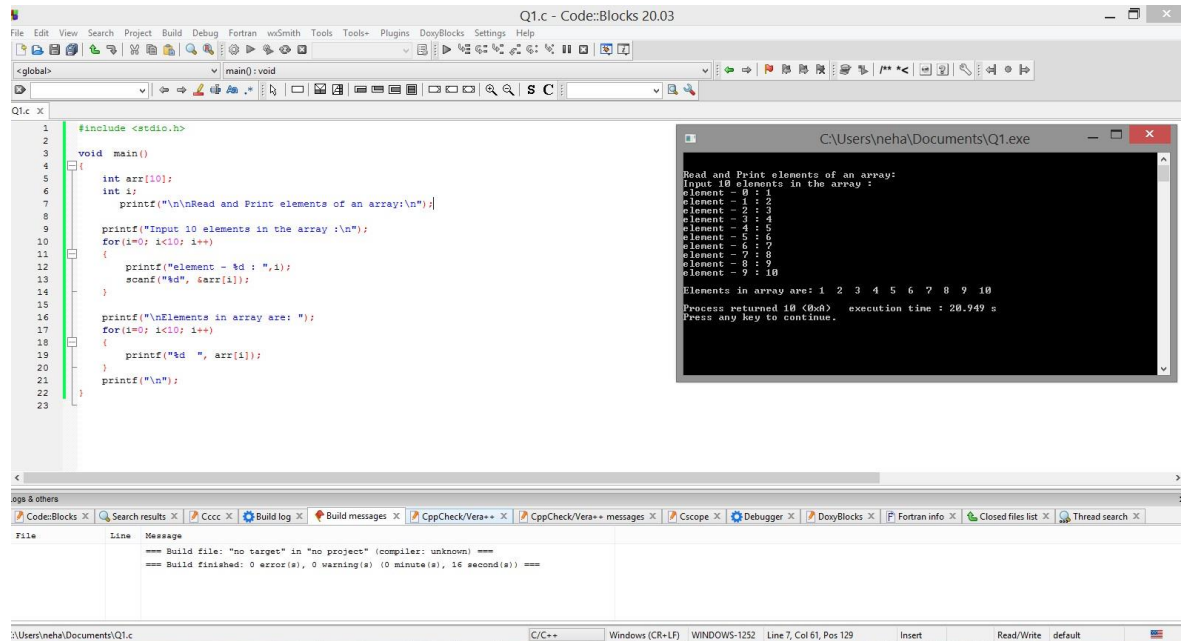
ASSIGNMENT 10

NAME: OM VIVEK GHARGE

PRN: 2020BTEIT00041

BRANCH: INFORMATION TECHNOLOGY

1) Write a program in C to store elements in an array and print it.



The screenshot displays the Code::Blocks IDE with a C program named Q1.c. The program is designed to read 10 elements into an array and then print them. The code is as follows:

```
1 #include <stdio.h>
2
3 void main()
4 {
5     int arr[10];
6     int i;
7     printf("\n\nRead and Print elements of an array:\n");
8
9     printf("Input 10 elements in the array :\n");
10    for(i=0; i<10; i++)
11    {
12        printf("element - %d : ", i);
13        scanf("%d", &arr[i]);
14    }
15
16    printf("\nElements in array are: ");
17    for(i=0; i<10; i++)
18    {
19        printf("%d ", arr[i]);
20    }
21    printf("\n");
22 }
23
```

The execution output, shown in a separate window titled "C:\Users\neha\Documents\Q1.exe", is as follows:

```
Read and Print elements of an array:
Input 10 elements in the array :
element - 0 : 1
element - 1 : 2
element - 2 : 3
element - 3 : 4
element - 4 : 5
element - 5 : 6
element - 6 : 7
element - 7 : 8
element - 8 : 9
element - 9 : 10

Elements in array are: 1 2 3 4 5 6 7 8 9 10
Process returned 16 (0x0)   execution time : 20.949 s
Press any key to continue.
```

The IDE's status bar at the bottom indicates the file path as "C:\Users\neha\Documents\Q1.c" and the current cursor position as "Line 7, Col 61, Pos 129".

2) Write a program in C to merge two arrays of same size sorted in descending order.

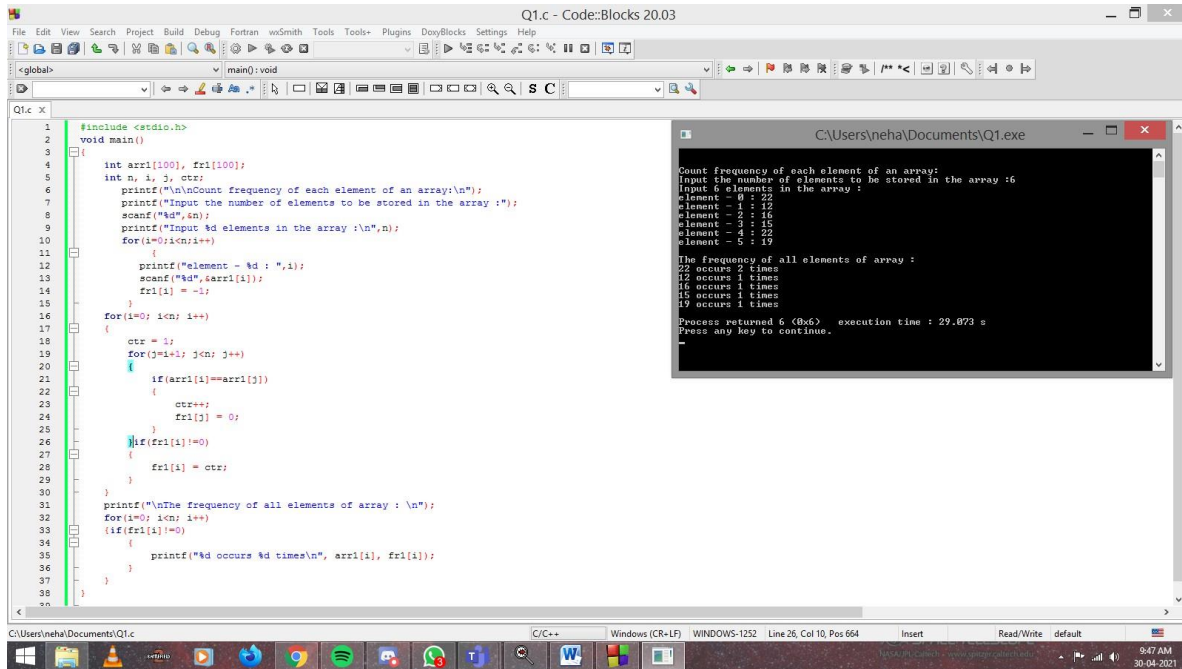
The screenshot shows a C code editor with the following code:

```
1 #include <stdio.h>
2 void main()
3 {
4     int arr1[100], arr2[100], arr3[200];
5     int s1, s2, s3;
6     printf("\nMerge two arrays of same size sorted in descending order.\n");
7     printf("Input the number of elements to be stored in the first array :");
8     scanf("%d", &s1);
9     printf("Input %d elements in the array :\n", s1);
10    for(i=0; i<s1; i++)
11        (printf("element - %d : ", i));
12    scanf("%d", &arr1[i]);
13    printf("Input the number of elements to be stored in the second array :");
14    scanf("%d", &s2);
15    printf("Input %d elements in the array :\n", s2);
16    for(i=0; i<s2; i++)
17        (printf("element - %d : ", i));
18    scanf("%d", &arr2[i]);
19    s3 = s1 + s2;
20    for(i=0; i<s3; i++)
21        (arr3[i] = arr1[i]);
22    for(j=0; j<s2; j++)
23        (arr3[s1+j] = arr2[j]);
24    for(i=0; i<s3; i++)
25        {
26            for(k=0; k<s3-1; k++)
27            {
28                if(arr3[k]<arr3[k+1])
29                {
30                    j=arr3[k+1];
31                    arr3[k+1]=arr3[k];
32                    arr3[k]=j;
33                }
34            }
35        }
36    printf("\nThe merged array in descending order is :\n");
37    for(i=0; i<s3; i++)
38        (printf("%d ", arr3[i]));
39    printf("\n\n");
40 }
```

The terminal window shows the following output:

```
C:\Users\neha\Documents\Q1.exe
Merge two arrays of same size sorted in descending order.
Input the number of elements to be stored in the first array : 5
Input 5 elements in the array :
element - 0 : 22
element - 1 : 56
element - 2 : 45
element - 3 : 78
element - 4 : 99
Input the number of elements to be stored in the second array : 5
Input 5 elements in the array :
element - 0 : 12
element - 1 : 43
element - 2 : 18
element - 3 : 97
element - 4 : 62
The merged array in descending order is :
99 97 78 62 56 45 43 22 18 12
Process returned 0 (0x0)   execution time : 33.328 s
Press any key to continue.
```

3) Write a program in C to count the frequency of each element of an array.



The screenshot shows the Code::Blocks IDE with a C program open in the editor and its output in a separate window.

Code::Blocks IDE - Q1.c

```
1 #include <stdio.h>
2 void main()
3 {
4     int arr1[100], frr1[100];
5     int n, i, j, ctr;
6     printf("\n\nCount frequency of each element of an array:\n");
7     printf("Input the number of elements to be stored in the array :");
8     scanf("%d", &n);
9     printf("Input %d elements in the array :\n", n);
10    for(i=0; i<n; i++)
11    {
12        printf("element - %d : ", i);
13        scanf("%d", &arr1[i]);
14        frr1[i] = -1;
15    }
16    for(i=0; i<n; i++)
17    {
18        ctr = 1;
19        for(j=i+1; j<n; j++)
20        {
21            if(arr1[i]==arr1[j])
22            {
23                ctr++;
24                frr1[j] = 0;
25            }
26            if(frr1[i]!=0)
27            {
28                frr1[i] = ctr;
29            }
30        }
31        printf("\nThe frequency of all elements of array : \n");
32        for(i=0; i<n; i++)
33        {
34            if(frr1[i]!=0)
35            {
36                printf("%d occurs %d times\n", arr1[i], frr1[i]);
37            }
38        }
39    }
```

Output Window: C:\Users\neha\Documents\Q1.exe

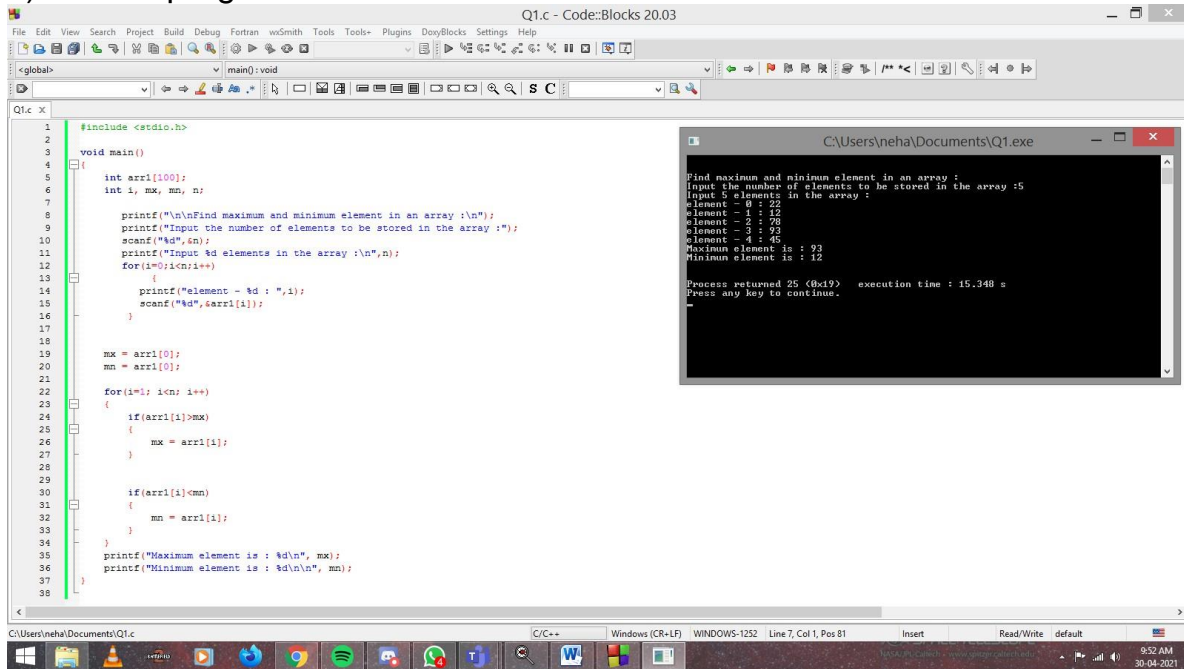
```
Count frequency of each element of an array:
Input the number of elements to be stored in the array :6
Input 6 elements in the array :
element - 0 : 22
element - 1 : 12
element - 2 : 16
element - 3 : 15
element - 4 : 22
element - 5 : 19

The frequency of all elements of array :
22 occurs 2 times
12 occurs 1 times
16 occurs 1 times
15 occurs 1 times
19 occurs 1 times

Process returned 0 (0x0), execution time : 29.073 s
Press any key to continue.
```

The IDE status bar at the bottom shows: C:\Users\neha\Documents\Q1.c, C/C++, Windows (CR-LF), WINDOWS-1252, Line 26, Col 10, Pos 664. The system tray at the bottom right shows the time as 9:47 AM on 30-04-2021.

4) Write a program in C to find the maximum and minimum element in an array.

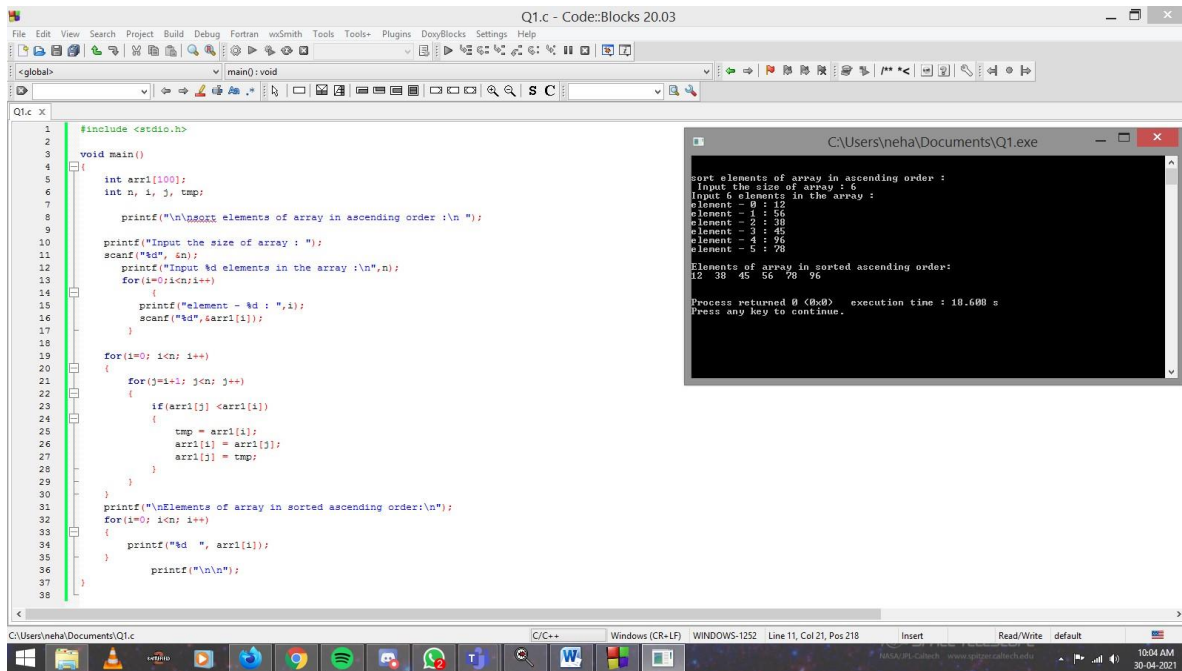


```
1 #include <stdio.h>
2
3 void main()
4 {
5     int arr1[100];
6     int i, mx, mn, n;
7
8     printf("\n\nFind maximum and minimum element in an array :\n");
9     printf("Input the number of elements to be stored in the array :");
10    scanf("%d",&n);
11    printf("Input %d elements in the array :\n",n);
12    for(i=0;i<n;i++)
13    {
14        printf("element - %d : ",i);
15        scanf("%d",&arr1[i]);
16    }
17
18    mx = arr1[0];
19    mn = arr1[0];
20
21    for(i=1; i<n; i++)
22    {
23        if(arr1[i]>mx)
24        {
25            mx = arr1[i];
26        }
27
28        if(arr1[i]<mn)
29        {
30            mn = arr1[i];
31        }
32    }
33
34    printf("Maximum element is : %d\n", mx);
35    printf("Minimum element is : %d\n", mn);
36
37 }
```

Find maximum and minimum element in an array :
Input the number of element to be stored in the array :5
Input 5 elements in the array :
element - 0 : 22
element - 1 : 12
element - 2 : 78
element - 3 : 93
element - 4 : 45
Maximum element is : 93
Minimum element is : 12

Process returned 25 (0x19) execution time : 15.348 s
Press any key to continue.

5) Write a program in C to sort elements of array in ascending order.



The screenshot shows a C program in Code::Blocks 20.03. The program is named Q1.c and is located in the directory C:\Users\neha\Documents. The code implements a sorting algorithm (likely bubble sort) to sort an array of 6 elements in ascending order. The output window shows the execution results, including the input array, the sorted array, and the execution time.

```
#include <stdio.h>

void main()
{
    int arr1[100];
    int n, i, j, tmp;

    printf("\n\nsort elements of array in ascending order :\n\n");

    printf("Input the size of array : ");
    scanf("%d", &n);
    printf("Input %d elements in the array :\n", n);
    for(i=0; i<n; i++)
    {
        printf("Element - %d : ", i);
        scanf("%d", &arr1[i]);
    }

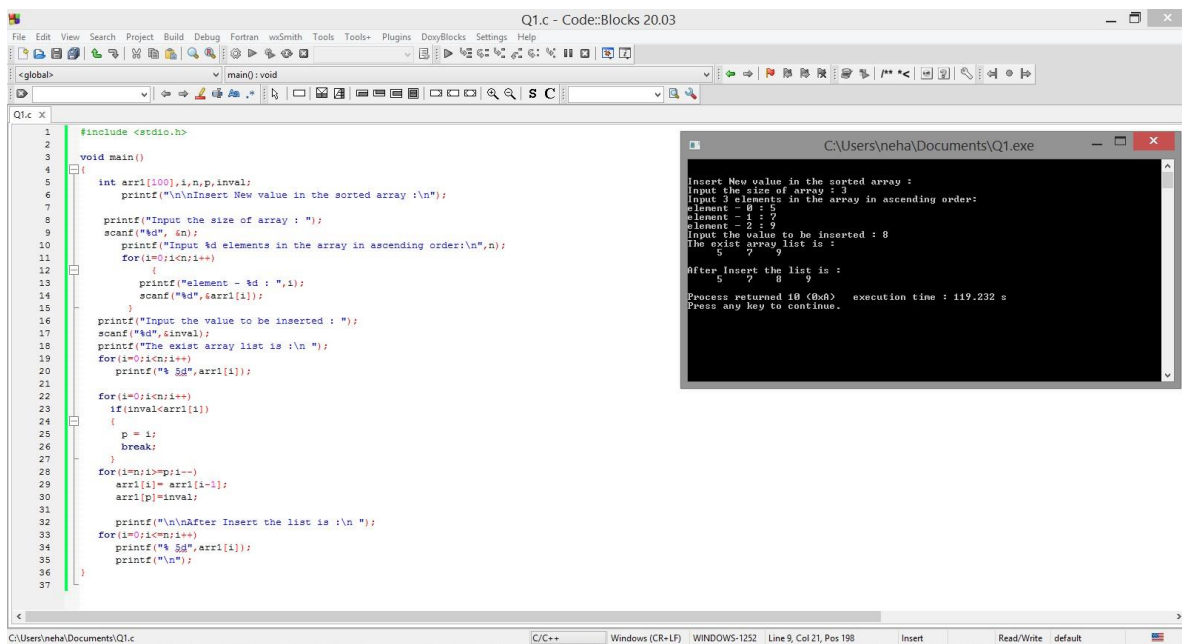
    for(i=0; i<n; i++)
    {
        for(j=i+1; j<n; j++)
        {
            if(arr1[j] < arr1[i])
            {
                tmp = arr1[i];
                arr1[i] = arr1[j];
                arr1[j] = tmp;
            }
        }
    }

    printf("\nElements of array in sorted ascending order:\n");
    for(i=0; i<n; i++)
    {
        printf("%d ", arr1[i]);
    }

    printf("\n\n");
}
```

sort elements of array in ascending order :
Input the size of array : 6
Input 6 elements in the array :
element - 0 : 12
element - 1 : 56
element - 2 : 38
element - 3 : 45
element - 4 : 96
element - 5 : 78
Elements of array in sorted ascending order:
12 38 45 56 78 96
Process returned 0 (0x0) execution time : 10.600 s
Press any key to continue.

6) Write a program in C to insert New value in the array (sorted list)
.. */int main(int argc, char const *argv[])



```
1 #include <stdio.h>
2
3 void main()
4 {
5     int arr1[100], i, n, p, inval;
6     printf("\n\nInsert New value in the sorted array :\n");
7
8     printf("Input the size of array : ");
9     scanf("%d", &n);
10    printf("Input %d elements in the array in ascending order:\n", n);
11    for(i=0; i<n; i++)
12    {
13        printf("element - %d : ", i);
14        scanf("%d", &arr1[i]);
15    }
16    printf("Input the value to be inserted : ");
17    scanf("%d", &inval);
18    printf("The exist array list is :\n ");
19    for(i=0; i<n; i++)
20        printf("%d\t", arr1[i]);
21
22    for(i=0; i<n; i++)
23        if(inval<arr1[i])
24        {
25            p = i;
26            break;
27        }
28    for(i=n; i>=p; i--)
29        arr1[i] = arr1[i-1];
30    arr1[p] = inval;
31
32    printf("\n\nAfter Insert the list is :\n ");
33    for(i=0; i<=p; i++)
34        printf("%d\t", arr1[i]);
35    printf("\n");
36 }
37
```

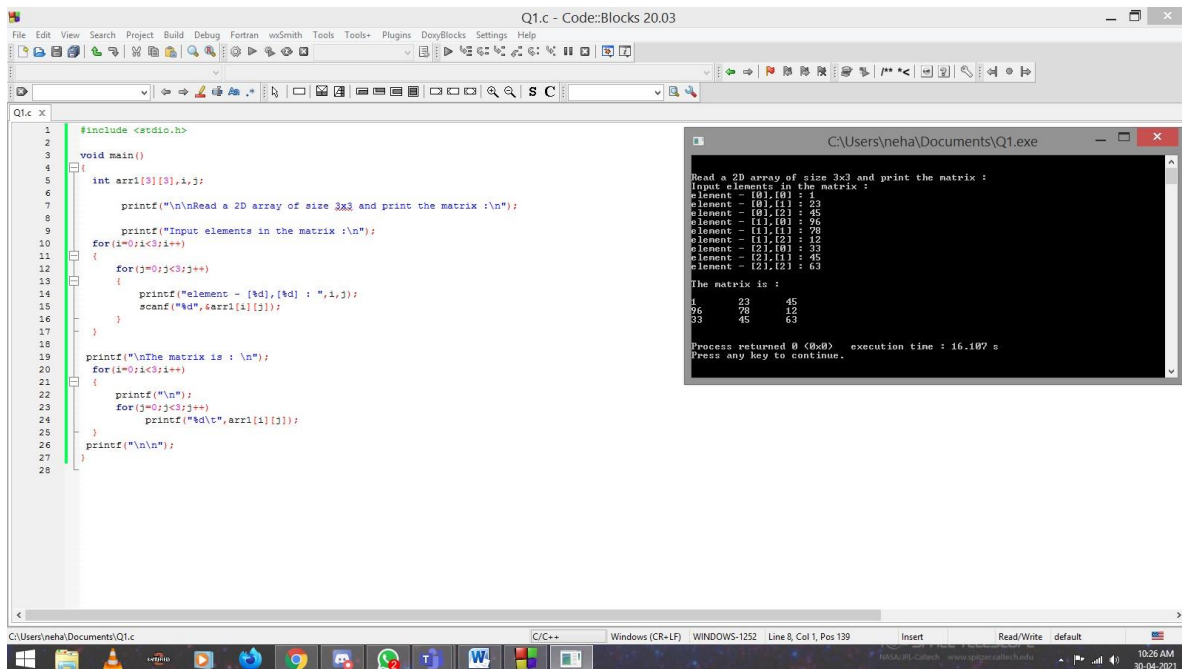
Output window (C:\Users\neha\Documents\Q1.exe):

```
Insert New value in the sorted array :
Input the size of array : 3
Input 3 elements in the array in ascending order:
element - 0 : 5
element - 1 : 7
element - 2 : 9
Input the value to be inserted : 8
The exist array list is :
5       7       9

After Insert the list is :
5       7       8       9

Process returned 10 (0x0A)   execution time : 119.232 s
Press any key to continue.
```

7) Write a program in C for a 2D array of size 3x3 and print the matrix.



```
1 #include <stdio.h>
2
3 void main()
4 {
5     int arr1[3][3], i, j;
6
7     printf("\n\nRead a 2D array of size 3x3 and print the matrix :\n");
8
9     printf("Input elements in the matrix :\n");
10    for(i=0; i<3; i++)
11    {
12        for(j=0; j<3; j++)
13        {
14            printf("element - [%d], [%d] : ", i, j);
15            scanf("%d", &arr1[i][j]);
16        }
17    }
18
19    printf("\n\nThe matrix is : \n");
20    for(i=0; i<3; i++)
21    {
22        printf("\n");
23        for(j=0; j<3; j++)
24        {
25            printf("%d\t", arr1[i][j]);
26        }
27        printf("\n\n");
28    }
```

Read a 2D array of size 3x3 and print the matrix :
Input elements in the matrix :
element - [0], [0] : 1
element - [0], [1] : 23
element - [0], [2] : 45
element - [1], [0] : 96
element - [1], [1] : 78
element - [1], [2] : 12
element - [2], [0] : 33
element - [2], [1] : 45
element - [2], [2] : 63

The matrix is :
1 23 45
96 78 12
33 45 63

Process returned 0 (0x0) execution time : 16.107 s
Press any key to continue.

8) Write a program in C for Addition of two matrices of same size.

The image shows a screenshot of a C program for adding two matrices of the same size. The program is written in Code::Blocks 20.03 and is titled "Q1.c". The source code is as follows:

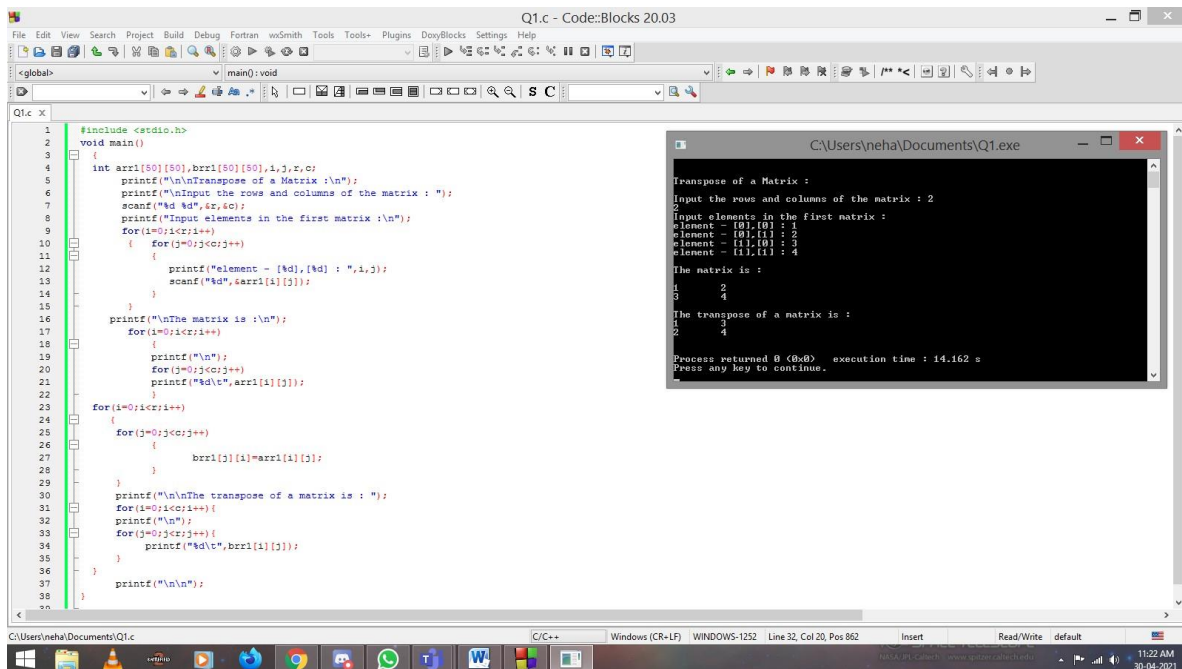
```
1 #include <stdio.h>
2 void main()
3 {
4     int arr1[50], arr2[50], arr3[50], i, j, n;
5     printf("\n\nAddition of two Matrices :\n\n");
6     scanf("%d", &n);
7     printf("Input the size of the square matrix (less than 5): ");
8     for(i=0; i<n; i++)
9     {
10         printf("element - [%d], [%d] : ", i, j);
11         scanf("%d", &arr1[i][j]);
12     }
13     printf("Input elements in the second matrix :\n\n");
14     for(i=0; i<n; i++)
15     {
16         for(j=0; j<n; j++)
17         {
18             printf("element - [%d], [%d] : ", i, j);
19             scanf("%d", &arr2[i][j]);
20         }
21     }
22     printf("\n\nThe First matrix is :\n\n");
23     for(i=0; i<n; i++)
24     {
25         printf("%d\n", arr1[i][0]);
26     }
27     printf("\n\nThe Second matrix is :\n\n");
28     for(i=0; i<n; i++)
29     {
30         printf("%d\n", arr2[i][0]);
31     }
32     printf("\n\nThe Addition of two matrix is : \n\n");
33     for(i=0; i<n; i++)
34     {
35         for(j=0; j<n; j++)
36         {
37             arr3[i][j] = arr1[i][j] + arr2[i][j];
38         }
39     }
40     printf("\n\nThe Addition of two matrix is : \n\n");
41     for(i=0; i<n; i++)
42     {
43         for(j=0; j<n; j++)
44         {
45             printf("%d\n", arr3[i][j]);
46         }
47     }
48 }
```

The output of the program is shown in a terminal window titled "C:\Users\neha\Documents\Q1.exe". The output is as follows:

```

Addition of two Matrices :
Input the size of the square matrix (less than 5): 2
Input elements in the first matrix :
element - [0], [0] : 4
element - [0], [1] : 5
element - [1], [0] : 6
element - [1], [1] : 7
Input elements in the second matrix :
element - [0], [0] : 1
element - [0], [1] : 2
element - [1], [0] : 3
element - [1], [1] : 4
The First matrix is :
4 5
6 7
The Second matrix is :
1 2
3 4
The Addition of two matrix is :
5 7
9 11
Process returned 0 (0x0)   execution time : 14.840 s
Press any key to continue.
```


9) Write a program in C to find transpose of a given matrix.



```
1 #include <stdio.h>
2 void main()
3 {
4     int arr1[50][50], brr1[50][50], i, j, r, c;
5     printf("\n\nTranspose of a Matrix : \n\n");
6     printf("\nInput the rows and columns of the matrix : ");
7     scanf("%d %d", &r, &c);
8     printf("Input elements in the first matrix : \n\n");
9     for(i=0; i<r; i++)
10     {
11         for(j=0; j<c; j++)
12         {
13             printf("element - [%d], [%d] : ", i, j);
14             scanf("%d", &arr1[i][j]);
15         }
16     }
17     printf("\nThe matrix is : \n\n");
18     for(i=0; i<r; i++)
19     {
20         printf("\n");
21         for(j=0; j<c; j++)
22             printf("%d\t", arr1[i][j]);
23     }
24     for(i=0; i<r; i++)
25     {
26         for(j=0; j<c; j++)
27         {
28             brr1[j][i] = arr1[i][j];
29         }
30     }
31     printf("\n\nThe transpose of a matrix is : ");
32     for(i=0; i<c; i++)
33     {
34         printf("\n");
35         for(j=0; j<r; j++)
36             printf("%d\t", brr1[i][j]);
37     }
38     printf("\n\n");
39 }
```

Output of the program:

```
Transpose of a Matrix :
Input the rows and columns of the matrix : 2
Input elements in the first matrix :
element - [0], [0] : 1
element - [0], [1] : 2
element - [1], [0] : 3
element - [1], [1] : 4

The matrix is :
1 2
3 4

The transpose of a matrix is :
1 3
2 4

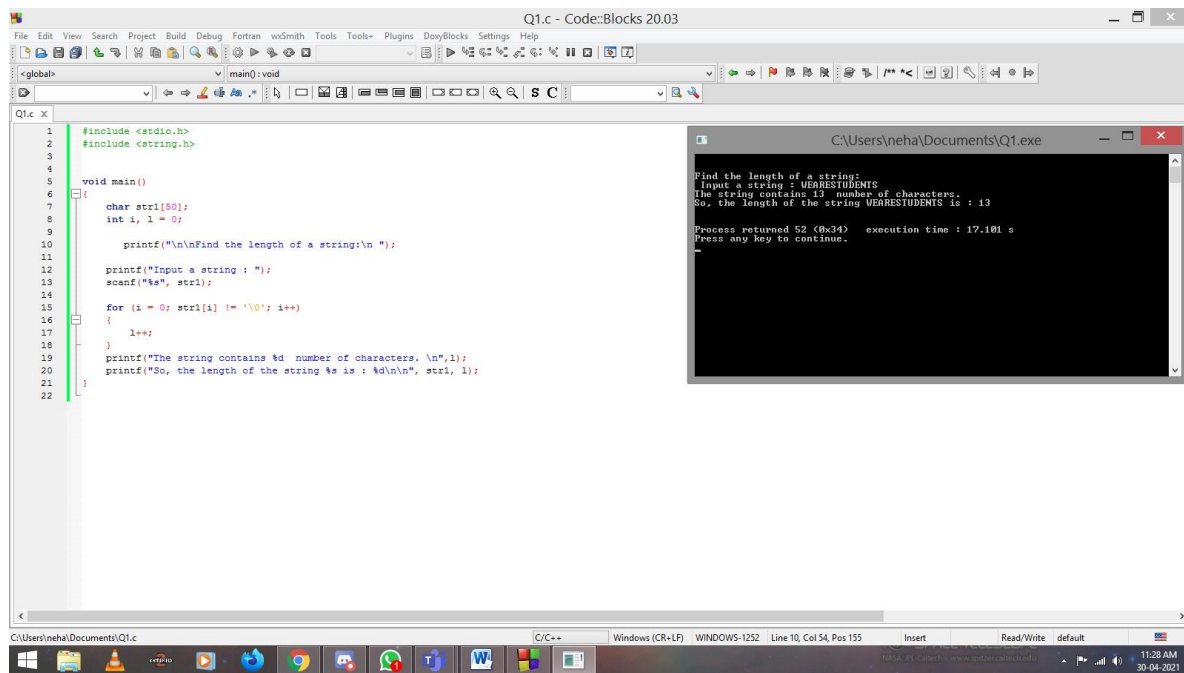
Process returned 0 (0x0)   execution time : 14.162 s
Press any key to continue.
```

10) Write a program in C to print all possible combinations of r elements in a given array.

```
1 #include <stdio.h>
2 void makeCombination(int arr1[], int data[], int st, int end,
3 int index, int r);
4 void CombinationDisplay(int arr1[], int n, int r)
5 {
6     int data[r];
7     makeCombination(arr1, data, 0, n-1, 0, r);
8 }
9 void makeCombination(int arr1[], int data[], int st, int end,
10 int index, int r)
11 {
12     if (index == r)
13     {
14         for (int j=0; j<r; j++)
15             printf("%d ", data[j]);
16         printf("\n");
17         return;
18     }
19     for (int i=st; i<=end && end-i+1 >= r-index; i++)
20     {
21         data[index] = arr1[i];
22         makeCombination(arr1, data, i+1, end, index+1, r);
23     }
24 }
25 int main()
26 {
27     int arr1[] = {1, 5, 4, 6, 8};
28     int r = 4;
29     int n = sizeof(arr1)/sizeof(arr1[0]);
30     printf("The given array is: \n");
31     for (i = 0; i < n; i++)
32     {
33         printf("%d ", arr1[i]);
34     }
35     printf("\n");
36     printf("The combination from by the number of elements are: %d\n", r);
37     printf("The combinations are: \n");
38     CombinationDisplay(arr1, n, r);
39 }
```

The given array is:
1 5 4 6 8
The combination from by the number of elements are: 4
The combinations are:
1 5 4 6
1 5 4 8
1 5 6 8
1 4 6 8
5 4 6 8
Process returned 0 (0x0) execution time : 0.024 s
Press any key to continue.

11) Write a program in C to find the length of a string with and without using library function.

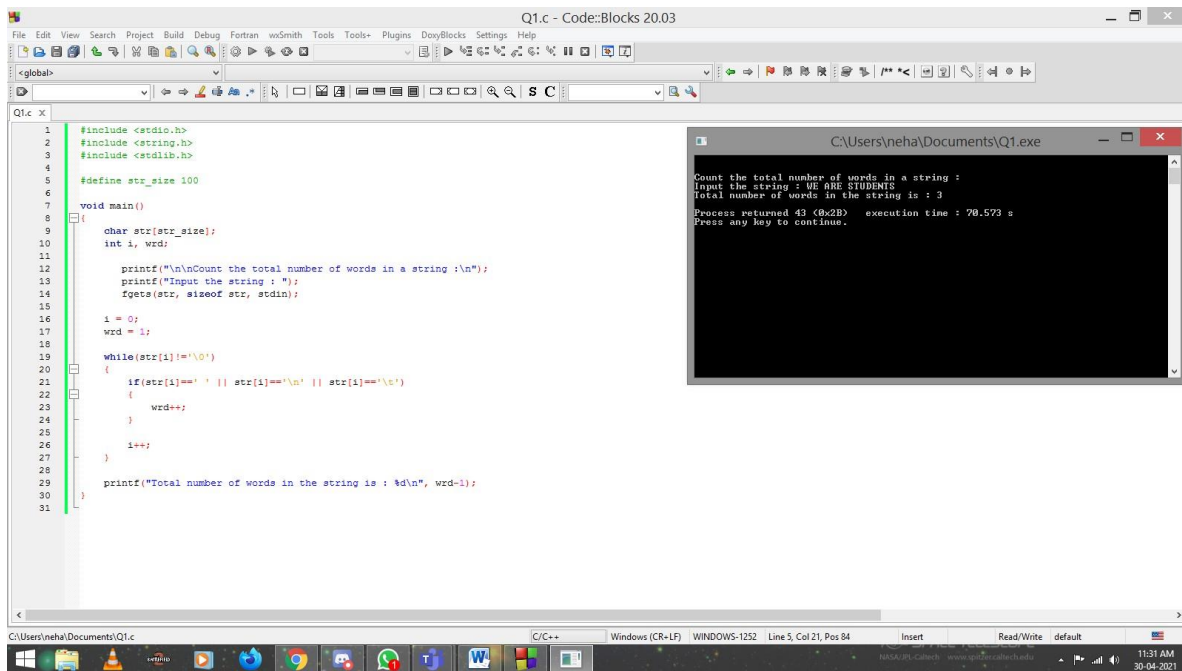


```
1 #include <stdio.h>
2 #include <string.h>
3
4
5 void main()
6 {
7     char str1[50];
8     int i, l = 0;
9
10    printf("\n\nFind the length of a string:\n ");
11
12    printf("Input a string : ");
13    scanf("%s", str1);
14
15    for (i = 0; str1[i] != '\0'; i++)
16    {
17        l++;
18    }
19    printf("The string contains %d number of characters. \n", l);
20    printf("So, the length of the string %s is : %d\n\n", str1, l);
21
22 }
```

Find the length of a string:
Input a string : WEARESTUDENTS
The string contains 13 number of characters.
So, the length of the string WEARESTUDENTS is : 13

Process returned 52 (0x34) execution time : 17.101 s
Press any key to continue.

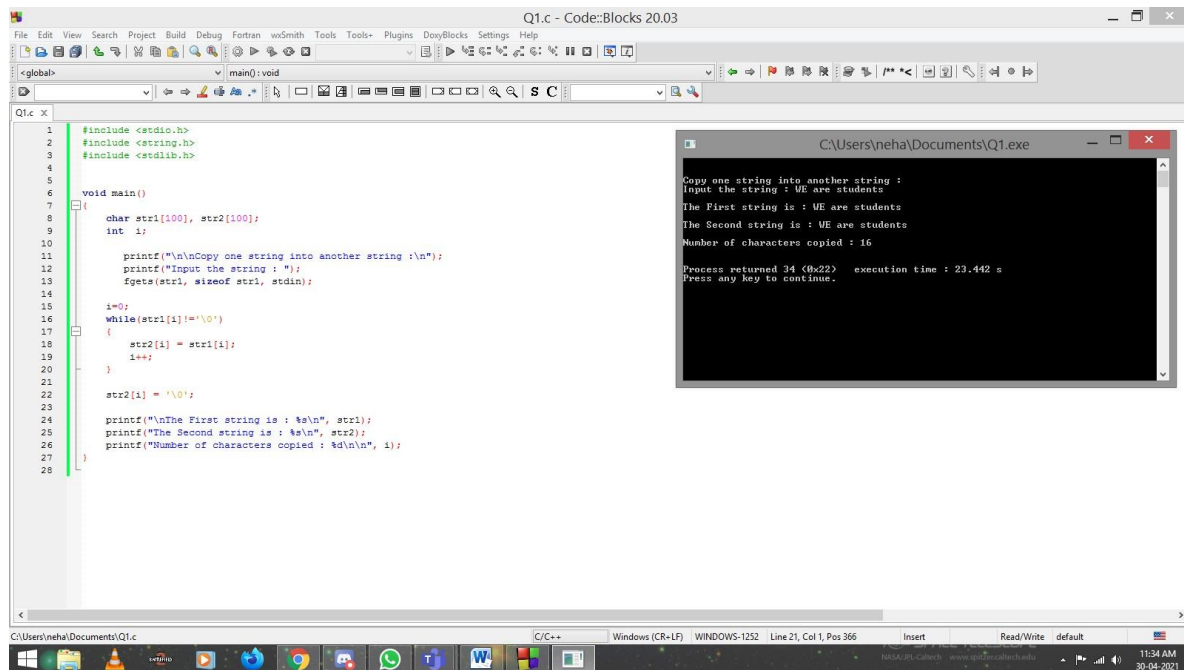
12) Write a program in C to count the total number of words in a string.



```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5 #define str_size 100
6
7 void main()
8 {
9     char str[str_size];
10    int i, wrd;
11
12    printf("\n\nCount the total number of words in a string :\n");
13    printf("Input the string : ");
14    fgets(str, sizeof str, stdin);
15
16    i = 0;
17    wrd = 1;
18
19    while(str[i]!='\0')
20    {
21        if(str[i]==' ' || str[i]=='\n' || str[i]=='\t')
22        {
23            wrd++;
24        }
25        i++;
26    }
27
28    printf("Total number of words in the string is : %d\n", wrd-1);
29
30 }
```

Count the total number of words in a string :
Input the string : WE ARE STUDENTS
Total number of words in the string is : 3
Process returned 43 (0x2B) execution time : 70.573 s
Press any key to continue.

13) Write a program in C to copy one string to another string.



The screenshot displays the Code::Blocks IDE with a C program for string copying. The program includes `<stdio.h>`, `<string.h>`, and `<stdlib.h>`. It defines a `main` function where two character arrays, `str1` and `str2`, are declared. A loop copies characters from `str1` to `str2` until a null terminator is reached. After copying, the program prints the first string, the second string, and the number of characters copied. The execution output window shows the program's runtime, including the input string "WE are students" and the resulting copied string.

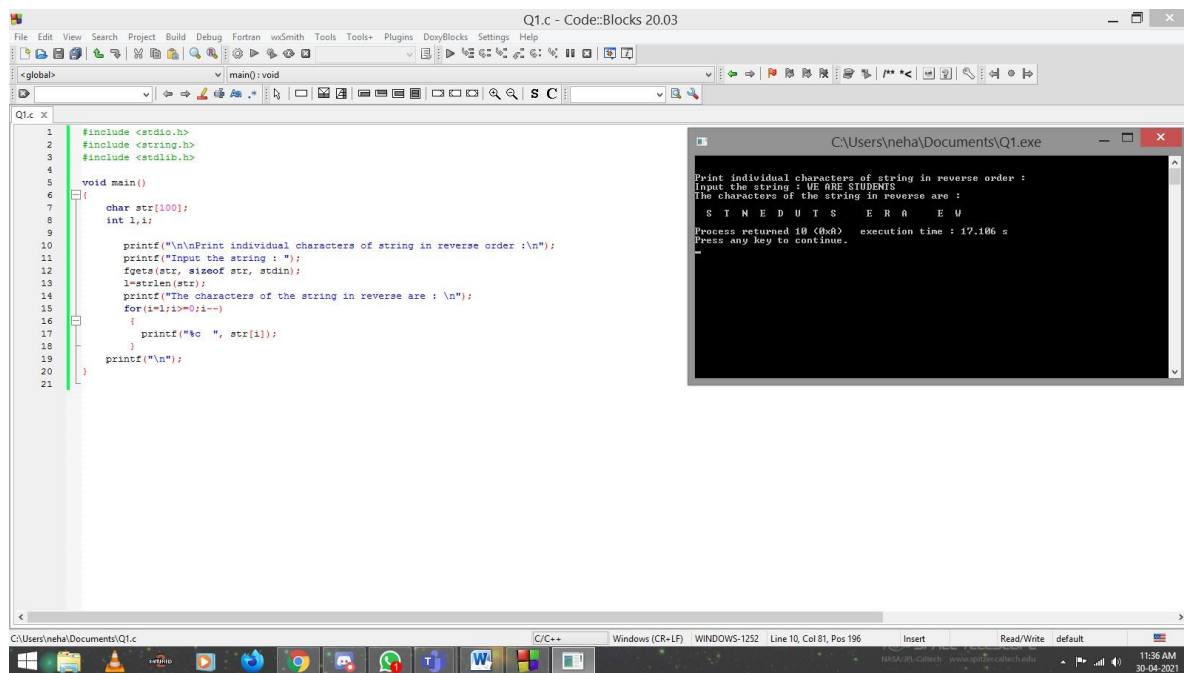
```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5
6 void main()
7 {
8     char str1[100], str2[100];
9     int i;
10
11     printf("\n\nCopy one string into another string :\n");
12     printf("Input the string : ");
13     fgets(str1, sizeof str1, stdin);
14
15     i=0;
16     while(str1[i]!='\0')
17     {
18         str2[i] = str1[i];
19         i++;
20     }
21
22     str2[i] = '\0';
23
24     printf("\nThe First string is : %s\n", str1);
25     printf("The Second string is : %s\n", str2);
26     printf("Number of characters copied : %d\n\n", i);
27 }
28
```

Output:

```
Copy one string into another string :
Input the string : WE are students
The First string is : WE are students
The Second string is : WE are students
Number of characters copied : 16

Process returned 34 (0x22)   execution time : 23.442 s
Press any key to continue.
```

14) Write a program in C to print individual characters of string in reverse order.

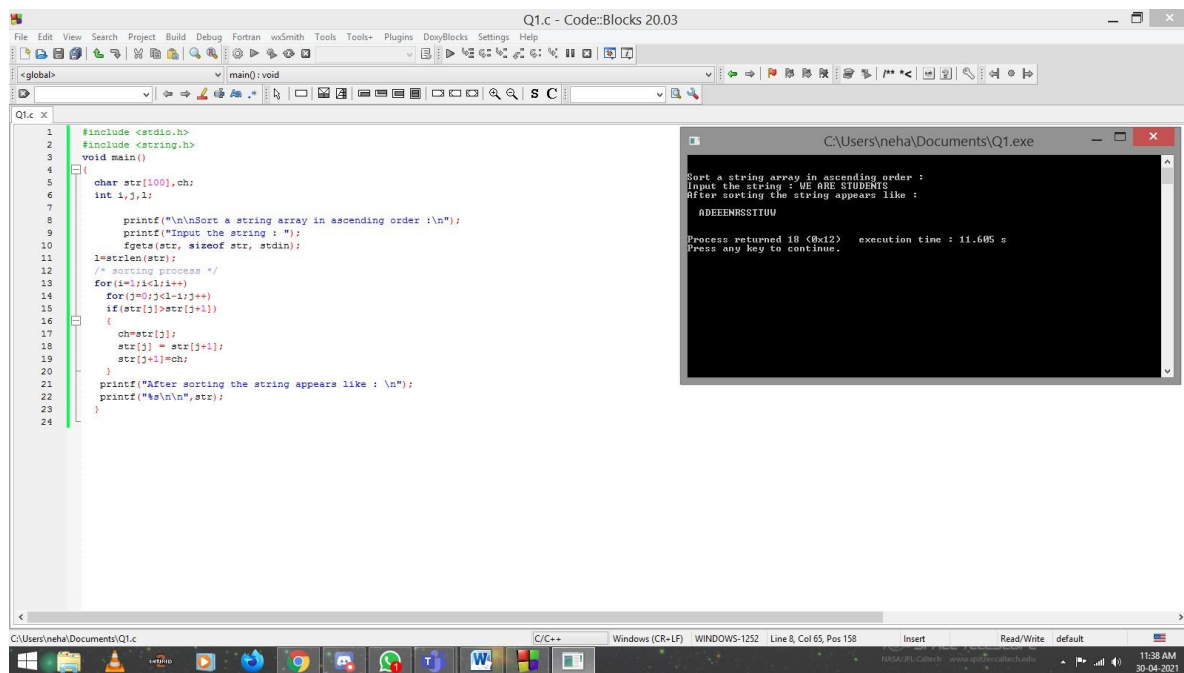


The screenshot displays the Code::Blocks 20.03 IDE. The main editor window shows a C program named Q1.c. The program includes headers for stdio.h, string.h, and stdlib.h. It defines a main function that declares a character array 'str' of size 100 and an integer 'i'. The program prompts the user to input a string, reads it using 'fgets', and then prints the characters in reverse order using a loop from the end of the string to the beginning. The output window shows the execution results, including the input string 'UE ARE STUDENTS', the reversed string 'STNEDUTS ERA EU', and the execution time of 17.186 seconds.

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4
5 void main()
6 {
7     char str[100];
8     int i,i;
9
10    printf("\n\nPrint individual characters of string in reverse order :\n");
11    printf("Input the string : ");
12    fgets(str, sizeof str, stdin);
13    i=strlen(str);
14    printf("The characters of the string in reverse are : \n");
15    for(i=i-1;i>=0;i--)
16    {
17        printf("%c ", str[i]);
18    }
19    printf("\n");
20 }
21
```

Print individual characters of string in reverse order :
Input the string : UE ARE STUDENTS
The characters of the string in reverse are :
STNEDUTS ERA EU
Process returned 10 (0x0) execution time : 17.186 s
Press any key to continue.

15) Write a C program to sort a string array in ascending order.



The screenshot shows the Code::Blocks 20.03 IDE with a C program open in the editor and its execution output in a separate window.

Code::Blocks 20.03 - Q1.c

```
1 #include <stdio.h>
2 #include <string.h>
3 void main()
4 {
5     char str[100],ch;
6     int i,j,l;
7
8     printf("\n\nSort a string array in ascending order :\n");
9     printf("Input the string : ");
10    fgets(str, sizeof str, stdin);
11    l=strlen(str);
12    /* sorting process */
13    for(i=0;i<l-1;i++)
14        for(j=i+1;j<l;j++)
15            if(str[i]>str[j])
16            {
17                ch=str[i];
18                str[i]=str[j];
19                str[j]=ch;
20            }
21    printf("After sorting the string appears like : \n");
22    printf("%s\n",str);
23 }
24
```

Execution Output (C:\Users\neha\Documents\Q1.exe)

```
Sort a string array in ascending order :
Input the string : DE ARE STUDENTS
After sorting the string appears like :
ADEENRSTIUW

Process returned 18 (0x12)   execution time : 11.605 s
Press any key to continue.
```

The program implements a bubble sort algorithm to sort the string "DE ARE STUDENTS" in ascending order, resulting in "ADEENRSTIUW".