

## DAY 2 : PRACTICE PROBLEMS

### ARRAYS & STRINGS

Kala Surendra

1. C Program to Find C Program to calculate the largest two numbers in a given Array.

#### **Problem Description**

We have to write a program in C such that the program will read the elements of a one-dimensional array, then compares the elements and finds which are the largest two elements in a given array.

#### **Expected Input and Output**

##### **Finding Largest 2 numbers in an array with unique elements:**

If we are entering 5 elements ( $N = 5$ ), with array element values as 2,4,5,8 and 7 then,

**The FIRST LARGEST = 8**

**THE SECOND LARGEST = 7**

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a file named 'ssss.c' containing the following C code:

```
1 #include <stdio.h>
2 int main()
3 {
4     int num1, num2;
5     printf("Please Enter Two different values\n");
6     scanf("%d %d", &num1, &num2);
7     if(num1 > num2)
8     {
9         printf("%d is Largest\n", num1);
10    }
11    else if (num2 > num1)
12    {
13        printf("%d is Largest\n", num2);
14    }
15    else
16    {
17        printf("Both are Equal\n");
18    }
19    return 0;
20 }
```

On the right, the terminal window shows the output of the program. It prompts the user to enter two values, then prints the larger value as the result. Below the terminal, the status bar shows the line number (9), column (41), and other compilation details.

2. C Program finds second largest & smallest elements in an Array.

### Problem Description

The program will implement a one-dimensional array and sort the array in descending order. Then it finds the second largest and smallest element in an array and also find the average of these two array elements. Later it checks if the resultant average number is present in a given array. If found, display appropriate message.

The screenshot shows the Dev-C++ IDE interface. The left pane displays the source code for 'ssss.c':

```

1 #include <stdio.h>
2 int main( )
3 {
4     int a[20], b[20], n, sml=0, pos, i, j, temp ;
5     printf(" Enter the Numbers of terms: ");
6     scanf("%d ",&n);
7     printf("\n Enter the terms : \n");
8     for ( i=1 ; i<n ; i++)
9     {
10        scanf("%d ",&a[i]);
11        b[i] = a[i];
12    }
13    for ( i=1 ; i<n ; i++)
14    {
15        for ( j=1 ; j<n ; j++)
16        {
17            if ( a[i] <= a[j])
18            {
19                temp = a[i];
20                a[i] = a[j];
21                a[j] = temp;
22            }
23        }
24        printf("\n The Array Elements are: \n");
25        for ( i=1 ; i<n ; i++)
26        printf(" %d ",b[i]);
27    }
28    printf("\n Second Smallest Element is : %d",a[2]);
29    printf("\n Second Largest Element is : %d",a[n-1]);
30    return ( 0 );
}

```

The right pane shows the terminal window output:

```

Enter the Numbers of terms: 7
8

Enter the terms :
11
26
32
45
22
10
55

The Array Elements are:
8      11      26      32      45      22
10
Second Smallest Element is : 10
Second Largest Element is : 32
-----
Process exited after 33.24 seconds with return value 0
Press any key to continue . . .

```

The status bar at the bottom provides system information:

Line: 24 Col: 15 Sel: 0 Lines: 30 Length: 603 Insert Done parsing in 0.016 seconds

System icons include: 30°, Search, File Explorer, Task View, File, Settings, Start, Taskbar, Network, Battery, ENG IN, 09:29, 30-03-2023.

### 3. C Program to find the odd element given an array with only two different elements.

#### Problem Description

This C Program finds odd element in a given array with only two different element.

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a file named 'ssss.c' containing the following C code:

```
1 #include<stdio.h>
2 #include<conio.h>
3 int main()
4 {
5     int arr[10], i;
6     printf("Enter any 10 array elements: ");
7     for(i=0; i<10; i++)
8         scanf("%d", &arr[i]);
9     printf("\nOdd Array elements are:\n");
10    for(i=0; i<10; i++)
11    {
12        if(arr[i]%2!=0)
13        {
14            printf("%d ", arr[i]);
15        }
16    }
17    getch();
18    return 0;
19 }
```

To the right, a terminal window titled 'D:\sварооп\ssss.exe' shows the output of the program. It prompts the user to enter 10 array elements and then lists them. Below that, it prints "Odd Array elements are:" followed by the odd elements from the array.

At the bottom of the Dev-C++ interface, there is a taskbar with various icons and system status information.

4. C Program to delete an element in an Array by index or value.

**Method 1: (Delete Element by Index)** In this approach, we will use a loop to iterate through the array and delete the element from the array.

**Example:**

**Input:**

Size of array = 5

array = [8, 4, 9, 6, 2] Index of the element to be deleted is “3”

**Output:**

[8, 4, 9, 2]

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays 'ssss.c' with the following C code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main(void)
4 {
5     int i, n, index, arr[10];
6     printf("Enter the size of the array: ");
7     scanf("%d", &n);
8     printf("Enter the elements of the array: \n");
9     for (i = 0; i < n; i++)
10    {
11        printf("arr[%d] = ", i);
12        scanf("%d", &arr[i]);
13    }
14    printf("Enter the index of the element to be deleted: ");
15    scanf("%d", &index);
16    if (index >= n+1)
17    {
18        printf ("\n Deletion is not possible in the array.");
19    }
20    else
21    {
22        for (i = index; i < n - 1; i++)
23            arr[i] = arr[i + 1];
24        printf("The array after deleting the element is: ");
25        for (i = 0; i < n - 1; i++)
26            printf("%d ", arr[i]);
27        return 0;
28    }
29 }
```

To the right, the terminal window shows the program's output:

```
D:\swaroop\ssss.exe
Enter the size of the array: 5
Enter the elements of the array:
arr[0] = 8,4,9,2
arr[1] = arr[2] = arr[3] = arr[4] = Enter the index of the element to be deleted: The array after deleting the element is: -1 20 0 1
-----
Process exited after 119.5 seconds with return value 0
Press any key to continue . . .
```

The status bar at the bottom shows the date and time: 30-03-2023 10:03.

5. C Program to remove duplicate elements from an Array.

**Input Array:** 1,2,4,5,4,2,7,5

**Output:** Resultant Array after removing duplicates: 1,2,4,5,7

The screenshot shows the Dev-C++ IDE interface. The left pane displays the source code for 'ssss.c'. The right pane shows the terminal window output. The terminal window title is 'D:\swaroop\ssss.exe'. The output shows the user entering elements of the array, the program processing the input to remove duplicates, and finally printing the resulting array.

```
#include<stdlib.h>
int main(){
    int a[50],i,j,k, count = 0, dup[50], number;
    printf("Enter size of the array");
    scanf("%d",&number);
    printf("Enter Elements of the array:");
    for(i=0;i<number;i++){
        scanf("%d",&a[i]);
        dup[i] = -1;
    }
    printf("Entered element are:");
    for(i=0;i<number;i++){
        printf("%d ",a[i]);
    }
    for(i=0;i<number;i++){
        for(j = i+1; j < number; j++){
            if(a[i] == a[j]){
                for(k = j; k < number; k++){
                    a[k] = a[k+1];
                }
                j--;
                number--;
            }
        }
    }
    printf("After deleting the duplicate element the Array is:");
    for(i=0;i<number;i++){
        printf("%d ",a[i]);
    }
}
```

Enter Elements of the array:  
2  
4  
5  
4  
2  
7  
5  
Entered element are:1 2 4 5 4 2 7 5 After deleting the duplicate element the Array is:1 2 4 5 7  
Process exited after 83.56 seconds with return value 5  
Press any key to continue . . .

## 6. C Program to reverse an Array.

**Reversing an array** means substituting the last element in the first position and vice versa and doing such a thing for all elements of the array. For **example**, first element is swapped with last, second element is swapped by second last and so on.

Such arrays where the original and reversed arrays are equal are called palindrome arrays.

**Examples:**

**Input array:** [1,2,3,4]

**Reversed array:** [4,3,2,1]

**Input array:** [3,2,1]

**Reversed array:** [1,2,3]

The screenshot shows a Windows desktop environment with a taskbar at the bottom. The taskbar icons include the Start button, a search bar, and various application icons like Microsoft Edge, File Explorer, and Google Chrome. There are four tabs open in the browser:

- Online C Compiler - c
- Perplexity AI
- C - Functions
- Introducing ChatGPT

The main window displays the "Online C Compiler" interface from CodingGround. The code editor contains the following C program:

```
#include <stdio.h>
void reverseArray(int arr[], int size) {
    int temp, i, j;
    for (i = 0, j = size - 1; i < j; i++, j--) {
        temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
}
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int size = sizeof(arr) / sizeof(arr[0]);
    int i;
    printf("Original array: ");
    for (i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    reverseArray(arr, size);
    printf("\nReversed array: ");
    for (i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

The terminal window on the right shows the output of the program:

```
Original array: 1
Reversed array: 1
```

7. Write a C program to merge two sorted array elements into a single array.

**Merging two arrays** in c is similar to Concatenating or combining two arrays into a single array. For **example**, if the first array has four elements and the second array has five elements, the resulting array has nine elements.

**Example:**

**First Array** = [1, 2, 3, 4, 5]

**Second Array** = [6, 7, 8, 9, 10]

**Merged Array** = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

### Problem Description

Write a C program to merge two sorted array elements into a single array.

The screenshot shows the Dev-C++ IDE interface. The left pane displays the source code for 'ssss.c' in the 'Editor' window. The right pane shows the 'D:\swaroop\ssss.exe' terminal window running the program. The terminal output shows the user input for the first array (size 5, elements 1-5) and the second array (size 5, elements 6-10), followed by the merged array (1-10). The status bar at the bottom indicates the process exited after 37.22 seconds.

```
1 #include <stdlib.h>
2 int main(void)
3 {
4     int i, n, j, k;
5     printf("Enter the size of the first array: ");
6     scanf("%d", &n);
7     int arr1[n];
8     printf("Enter the elements of the first array: \n");
9     for (i = 0; i < n; i++)
10    {
11        scanf("%d", &arr1[i]);
12    }
13    printf("Enter the size of the second array: ");
14    scanf("%d", &k);
15    int arr2[k];
16    printf("Enter the elements of the second array: \n");
17    for (j = 0; j < k; j++)
18    {
19        scanf("%d", &arr2[j]);
20    }
21    int arr3[n + k];
22    i = 0;
23    int in;
24    for (in = 0; in < n + k; in++)
25    {
26        if (i < n && j < k)
27        {
28            if (arr1[i] < arr2[j])
29            {
30                arr3[in] = arr1[i];
31                i++;
32            }
33        }
34    }
35 }
```

8. Write a program to find the number of composite numbers in an array of elements

**Sample Input:**

Array of elements = {16, 18, 27, 16, 23, 21, 19}

**Sample Output:**

Number of Composite Numbers = 5

**Test cases:**

1. Array of elements = {26, 28, 37, 26, 33, 31, 29}
2. Array of elements = {1.6, 1.8, 2.7, 1.6, 2.3, 2.1, .19}
3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}
4. Array of elements = {200, 180, 180, 270, 270, 270, 190, 200}
5. Array of elements = {100, 100, 100, 100, 100, 100, 100}

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a file named 'ssss.c' containing C code. The code prompts the user for the size of an array and its elements, then iterates through the array to count composite numbers. The right side shows the terminal window where the program is run, displaying the input values (size 5 and elements 26, 28, 37, 26, 33) and the output (total composite number are: 3). The status bar at the bottom provides build information and system details.

```
1 #include<stdio.h>
2 void main (){
3     int i,n,a[100],count=0;
4     printf("enter size:");
5     scanf("%d",&n);
6     printf("enter elements\n");
7     for(i=0;i<n;i++){
8         scanf("%d",&a[i]);
9     }
10    for(i=0;i<n;i++){
11        if(a[i]==2){
12            continue;
13        }
14        else if(a[i]>2&&a[i]<=2){
15            count++;
16        }
17    }
18    if(count>2){
19    }
20    printf("total composite number are: %d",count);
21 }
```

D:\swaroop\ssss.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

ssss.c (globals)

```
1 #include<stdio.h>
2 void main (){
3     int i,n,a[100],count=0;
4     printf("enter size:");
5     scanf("%d",&n);
6     printf("enter elements\n");
7     for(i=0;i<n;i++){
8         scanf("%d",&a[i]);
9     }
10    for(i=0;i<n;i++){
11        if(a[i]==2){
12            continue;
13        }
14        else if(a[i]>2&&a[i]<=2){
15            count++;
16        }
17    }
18    if(count>2){
19    }
20    printf("total composite number are: %d",count);
21 }
```

D:\swaroop\ssss.exe

```
enter size:5
enter elements
26
28
37
26
33
total composite number are: 3
Process exited after 25.49 seconds with return value 29
Press any key to continue . . .
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation Line: 1 Col: 18 Sel: 0 Lines: 21 Length: 448 Insert Done parsing in 0 seconds

28% Search ENG IN 01:05 30-03-2023

10. Write a program to print the number of negative numbers in an array of number

**Sample Input:**

Array of elements = {16, -18, 27, -16, 23, -21, 19}

**Sample Output:**

Number of negative numbers in Array elements = 3

**Test cases:**

1. Array of elements = {-26, 28, 37, -26, 33, -31, -29}

2. Array of elements = {1.6, 1.8, 2.7, -1.6, 2.3, -2.1, .19}

3. Array of elements = {0, 160, 180, 270, 160, 230, 210, 190, 0}

4. Array of elements = {-16, 2.8, -7, -1.5, 2.8, -2.8, -.19}

5. Array of elements = {-160, -160, -180, -270, -160, -230, -210, 1-90, 0}

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a file named 'ssss.c' containing the following C code:

```
1 #include<stdio.h>
2 int main()
3 {
4     int Size, i, a[10];
5     printf("\n Please Enter the Size of an Array : ");
6     scanf("%d", &Size);
7     printf("\n Please Enter the Array Elements : ");
8     for(i = 0; i < Size; i++)
9     {
10        |   scanf("%d", &a[i]);
11    }
12    printf("\n List of Negative Numbers in this Array : ");
13    for(i = 0; i < Size; i++)
14    {
15        if(a[i] < 0)
16        {
17            |   printf("%d ", a[i]);
18        }
19    }
20    return 0;
21 }
```

To the right, a terminal window titled 'D:\swaroop\ssss.exe' shows the program's output:

```
Please Enter the Size of an Array : 5
Please Enter the Array Elements : -26
28
37
-26
33

List of Negative Numbers in this Array : -26 -26
-----
Process exited after 41.32 seconds with return value 0
Press any key to continue . . .
```

At the bottom, the status bar shows the current line (Line 11), column (Col: 3), selected text (Sel: 0), lines (Lines: 21), length (Length: 422), and the message "Done parsing in 0.031 seconds". The taskbar at the very bottom includes icons for File Explorer, Task View, Search, and various system and application icons.

11. Write a program to search the given element and display its position in a linear array.

**Sample Input:**

Array of elements = { 16, 18, 27, 16, 23, 21, 19}

Element to search = 23

**Sample Output:**

Given element 23 is found at 5 th position

The screenshot shows the Dev-C++ IDE interface. The code editor window displays a C program named ssss.c. The code implements a linear search algorithm to find a key number in an array of integers. The user is prompted to enter the number of elements and the elements themselves, followed by the element to be searched. The program then iterates through the array to find the target value. If found, it prints the position; otherwise, it indicates the element is not present. The terminal window shows the execution of the program, including the user input and the program's output. The status bar at the bottom provides build information and system details.

```
#include <stdio.h>
void main()
{
    int num;
    int i, keynum, found = 0;
    printf("Enter the number of elements ");
    scanf("%d", &num);
    int array[num];
    printf("Enter the elements one by one \n");
    for (i = 0; i < num; i++)
    {
        scanf("%d", &array[i]);
    }
    printf("Enter the element to be searched ");
    scanf("%d", &keynum);
    for (i = 0; i < num; i++)
    {
        if (keynum == array[i])
        {
            found = 1;
            break;
        }
    }
    if (found == 1)
        printf("Element is present in the array at position %d", i+1);
    else
        printf("Element is not present in the array\n");
}
```

D:\swaroop\ssss.c - [Executing] - Dev-C++ 5.11  
File Edit Search View Project Execute Tools AStyle Window Help  
TDM-GCC 4.9.2 64-bit Release  
D:\swaroop\ssss.exe  
Enter the number of elements 7  
Enter the elements one by one  
16  
18  
27  
16  
23  
21  
19  
Enter the element to be searched 21  
Element is present in the array at position 6  
-----  
Process exited after 46.06 seconds with return value 45  
Press any key to continue . . . |  
Compiler Resources Compile Log Debug Find Results Close  
Abort Compilation Line: 27 Col: 2 Sel: 0 Lines: 27 Length: 682 Insert Done parsing in 0.016 seconds  
28° Search ENG IN 01:21 30-03-2023

12. Write a program to find the sum and average of the elements in an array

**Sample Input:**

Array of elements = {16, 18, 27, 16, 23, 21, 19}

**Sample Output:**

Sum = 140

Average = 20

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a C program named 'ssss.c' with syntax highlighting. The code prompts the user for the number of elements and their values, calculates the sum and average, and prints the results. On the right, a terminal window titled 'D:\swaroop\ssss.exe' shows the execution of the program, including user input and the program's output.

```
1 #include <stdio.h>
2 int main()
3 {
4     int Arr[100], n, i, sum = 0;
5     printf("Enter the number of elements you want to insert : ");
6     scanf("%d", &n);
7     for (i = 0; i < n; i++)
8     {
9         printf("Enter element %d : ", i + 1);
10        scanf("%d", &Arr[i]);
11        sum += Arr[i];
12    }
13    printf("\nThe sum of the array is : %d", sum);
14    printf("\nThe average of the array is : %.2f", (float)sum / n);
15    return 0;
16 }
```

D:\swaroop\ssss.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

SSSS.C

TDM-GCC 4.9.2 64-bit Release

D:\swaroop\ssss.exe

Enter the number of elements you want to insert :  
7  
Enter element 1 : 16  
Enter element 2 : 18  
Enter element 3 : 27  
Enter element 4 : 16  
Enter element 5 : 23  
Enter element 6 : 21  
Enter element 7 : 19

The sum of the array is : 140  
The average of the array is : 20.00

Process exited after 32.52 seconds with return value 0  
Press any key to continue . . .

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Line: 14 Col: 69 Sel: 0 Lines: 16 Length: 443 Insert Done parsing in 0.016 seconds

28° Search

01:29 30-03-2023 ENG IN

13. Write a Program to display the diagonal elements in a matrix array and also find the sum of them.

## Sample input:

123

456

789

## Output:

Diagonal Elements are 1 5 9

Sum of diagonal elements = 15

The screenshot shows a Dev-C++ IDE interface with the following details:

- Title Bar:** D:\swaroop\ssss.c - [Executing] - Dev-C++ 5.11
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Includes icons for file operations like Open, Save, Print, and Build.
- Project Explorer:** Shows the file ssss.c and its contents.
- Code Editor:** The code is as follows:

```
1 #include <stdio.h>
2 int main()
3 {
4     int i, j, m = 3, n = 3, a = 0, sum = 0;
5
6     // input matrix
7     int matrix[3][3]
8         = {{ 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };
9     if (m == n) {
10         printf("The matrix is \n");
11         for (i = 0; i < m; ++i) {
12             for (j = 0; j < n; ++j) {
13                 printf("%d", matrix[i][j]);
14             }
15             printf("\n");
16         }
17         for (i = 0; i < m; ++i) {
18             sum = sum + matrix[i][i];
19             a = a + matrix[i][m - i - 1];
20         }
21         printf("\nMain diagonal elements sum is = %d\n", sum);
22         printf("Off-diagonal elements sum is = %d\n", a);
23     }
24     else
25         printf("not a square matrix\n");
26     return 0;
27 }
```

- Output Window:** Displays the execution results of the program.
- Bottom Navigation:** Compiler, Resources, Compile Log, Debug, Find Results, Close.
- Status Bar:** Line: 1 Col: 19 Sel: 0 Lines: 28 Length: 607 Insert Done parsing in 0.015 seconds.
- Taskbar:** Shows various system icons and the current date and time (01:38, 30-03-2023).

14. Write a Program to find the Maximum and Minimum value in a given array of numbers.

## Sample Input:

Enter no. of elements in an array 5

Enter the elements:

1 2 3 4 5

## **Output:**

### Maximum of an array 5

### Minimum of an array 1

The screenshot shows the Dev-C++ IDE interface with a C program named ssss.c. The code reads an array of integers from the user and prints the minimum and maximum values. The execution window shows the program's output: it asks for the size of the array (5), then the elements (1, 2, 3, 4, 5), and finally prints the minimum (1) and maximum (5). The process exits after 29.18 seconds.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a[100], i, n, min, max;
    printf("Enter size of the array : ");
    scanf("%d", &n);
    printf("Enter elements in array : ");
    for (i=0; i<n; i++)
    {
        scanf("%d", &a[i]);
    }
    min=max=a[0];
    for (i=1; i<n; i++)
    {
        if(min>a[i])
            min=a[i];
        if(max<a[i])
            max=a[i];
    }
    printf("minimum of array is : %d",min);
    printf("\nmaximum of array is : %d",max);
    return 0;
}
```

15. Write a program to print the number of vowels in the given statement?

**Sample Input:**

Saveetha School of Engineering

**Sample Output:**

Number of vowels = 12

**Test cases:**

1. India is my country
2. All are my brothers and sisters
3. Why dry sky
4. Shy Try Cry
5. EDUCATION

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays 'ssss.c' with the following C code:

```
1 #include <stdio.h>
2 #include <ctype.h>
3 int isVowel(char ch)
4 {
5     ch = tolower(ch);
6     if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
7         return 1;
8     return 0;
9 }
10 int getVowelCount(char a[])
11 {
12     int count = 0, i = 0;
13     char ch;
14     while (a[i] != '\0'){
15         ch = a[i];
16
17         if (isVowel(ch))
18             count++;
19         i++;
20     }
21     return count;
22 }
23 int main()
24 {
25     char str[100];
26     printf("Enter a string\n");
27     scanf("%[^\n]",&str);
28     printf("Total Vowels: %d\n", getVowelCount(str));
29
30
31     return 0;
32 }
```

On the right, the terminal window shows the execution results:

```
D:\swaroop\ssss - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(ssss) (globals)

D:\swaroop\ssss.exe
Enter a string
saveetha school 0f engineering
Total Vowels: 11
-----
Process exited after 30.7 seconds with return
value 0
Press any key to continue . . . |
```

The status bar at the bottom provides system information: Line: 2 Col: 19 Sel: 0 Lines: 32 Length: 577 Insert Done parsing in 0.031 seconds. The taskbar at the bottom right shows various application icons.

16. Write a C program that accepts a string and checks whether a given string is a palindrome or not.

**Example 1:**

**Input:** String = “ABEEBA”

**Output:** ABEEBA is a Palindrome

**Explanation:** The reverse of the given string is equal to the (ABEEBA) is equal to the given string. Therefore, the given string is a palindromic string.

**Example 2:**

**Input:** String = “SANFOUNDRY”

**Output:** SANFOUNDRY is not a Palindrome

**Explanation:** The reverse of the given string is equal to the (YRDNUOFNAS) which is not equal to the given string. Therefore, the given string is not a palindromic string.

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a file named 'ssss.c' containing the following C code:

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char string1[20];
6     int i, length;
7     int flag = 0;
8     printf("Enter a string:");
9     scanf("%s", string1);
10    length = strlen(string1);
11    for(i=0;i < length ;i++){
12        if(string1[i] != string1[length-i-1]){
13            flag = 1;
14            break;
15        }
16    }
17    if (flag) {
18        printf("%s is not a palindrome", string1);
19    }
20    else {
21        printf("%s is a palindrome", string1);
22    }
23 }
```

To the right of the code editor is a terminal window titled 'D:\swaroop\ssss.exe' showing the program's output:

```
Enter a string:abeba
abeba is a palindrome
-----
Process exited after 8.783 seconds with return value 0
Press any key to continue . . . |
```

At the bottom of the Dev-C++ interface, there is a toolbar with various icons and a status bar displaying information such as 'Line: 3 Col: 4 Sel: 0 Lines: 23 Length: 521 Insert Done parsing in 0.016 seconds' and system status like '28°' and 'Search'.

17. Write a C program that takes two strings as input and concatenates them.

### Problem Solution

1. Take the two strings as input.
2. Concatenate the second string to the first string.
3. Print the concatenated string.

#### Input:

Enter the first string: “Sanfoundry”

Enter the second string: “Programming”

#### Output:

Concatenated string = SanfoundryProgramming

The screenshot shows the Dev-C++ IDE interface. The code editor window displays a file named 'ssss.c' containing the following C code:

```
1 #include <stdio.h>
2 #include<string.h>
3 int main()
4 {
5     char s1[20];
6     char s2[20];
7     printf("Enter the first string : ");
8     scanf("%s", s1);
9     printf("\nEnter the second string : ");
10    scanf("%s", s2);
11    strcat(s1,s2);
12    printf("The concatenated string is : %s",s1);
13    return 0;
14 }
```

To the right of the code editor is a terminal window titled 'D:\swaroop\ssss.exe' showing the execution results:

```
Enter the first string : sanfoundry
Enter the second string :programming
The concatenated string is : sanfoundryprogramming
-----
Process exited after 36.09 seconds with return value
0
Press any key to continue . . .
```

At the bottom of the Dev-C++ interface, there is a status bar displaying various system and application icons, along with the date and time (02:14 30-03-2023).

18. Write a C Program to compare two strings.

String is a sequence of characters terminated by the special character '\0'. Strings can be compared with or without using the string function.

**Example:**

String1="Hello"    String2="Hello"    Both string are equal

String1="Hello"    String2="Hell"    String1 is greater

String1="Hello"    String2="Helz"    String2 is greater

**Problem Solution**

1. Take two strings as input.
2. Compare the two strings and display the result whether both are equal, or first string is greater than the second or the first string is less than the second string.
3. Exit.

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a C program named 'ssss.c' with syntax highlighting. On the right, the terminal window shows the execution of the program. The user enters 'hello' for both strings, and the program outputs 'Both strings are EQUAL'. The terminal also shows the process exit information and a prompt to press any key to continue.

```
#include <stdio.h>
int main ()
{
    int count1 = 0, count2 = 0, flag = 0, i;
    char string1[30], string2[30];
    printf ("Enter the First string\n");
    gets (string1);
    printf ("Enter the Second string\n");
    gets (string2);
    while (string1[count1] != '\0')
        count1++;
    while (string2[count2] != '\0')
        count2++;
    i = 0;
    while (string1[i] == string2[i] && string1[i] != '\0')
    {
        i++;
    }
    if (string1[i] > string2[i])
        printf ("First string is greater than Second string\n");
    else if (string1[i] < string2[i])
        printf ("Second string is greater than First string\n");
    else
        printf ("Both strings are EQUAL\n");
    return 0;
}
```

D:\swaroop\ssss.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

D:\swaroop\ssss.exe

Enter the First string  
hello  
Enter the Second string  
hello  
Both strings are EQUAL

-----  
Process exited after 23.11 seconds with return value 0  
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Line: 24 Col: 45 Sel: 0 Lines: 26 Length: 655 Insert Done parsing in 0.016 seconds

28° Q Search

ENG IN 02:18 30-03-2023

19. Write a C Program to remove all characters in second string which are present in first string.

**Problem Description**

C Program removes all characters from second string which were present in the first string.

**Problem Solution**

Take input from the user and perform string operations as shown in the program below.

**Test case 1** – Here, the First & Second string are different.

Enter the First string:

Programming

Enter the Second string:

Computer

On removing characters from second string we get: Cpute

**Test case 2** – Here, the First & Second string are having some common letters.

Enter the First string:

sanfoundry

Enter the Second string:

sanppppdry

On removing characters from second string we get: pppp

```

D:\swaroop\ssss.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(ssss.c)
4 int main ()
5 {
6     char str[50], str1[50], rem[100];
7     int x = 0, y = 0, z = 0;
8     printf ("Enter The Main String:\n");
9     fflush (stdin);
10    gets (str);
11    printf ("Enter The Second String To Find:\n");
12    gets (str1);
13    for (x = 0; str[x] != '\0'; x++)
14    {
15        for (y = 0; str1[y] != '\0'; y++)
16        {
17            if (str[x] == str1[y])
18            {
19                continue;
20            }
21            else
22            {
23                rem[z] = str1[y];
24                z++;
25            }
26        }
27        rem[z] = '\0';
28        strcpy (str1, rem);
29        z = 0;
30    }
31    printf ("Removed The Characters From The Second String: %s\n", rem);
32    return 0;
33 }

```

D:\swaroop\ssss.exe

Enter The Main String:  
sunfoundry  
Enter The Second String To Find:  
sanpppdry  
Removed The Characters From The Second String:  
apppp

-----  
Process exited after 47.85 seconds with return value 0  
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Line: 31 Col: 73 Sel: 0 Lines: 33 Length: 773 Insert Done parsing in 0.016 seconds

28° Search

28° Search

20. This is a C Program to replace all the characters by lowercase.

### Problem Description

This is a C Program to replace all the characters by lowercase.

### Problem Solution

1. Take a string as input.
2. Use string function strlwr to replace all the characters by lowercase.

### Input a string to convert to lower case

CHANDANA chanikya  
rAVELLA

### Input string in lower case:

chandana chanikya  
ravella

The screenshot shows the Dev-C++ IDE interface. The left pane displays the source code for a C program named `ssss.c`. The code includes `#include <stdio.h>`, `#include <string.h>`, and a `main()` function that prompts for input, converts it to lowercase using `strlwr()`, and prints the result. The right pane shows the terminal window output. The user inputs "chandana chanikya", and the program outputs "chandana chanikya" in lowercase. The terminal also shows the process exit information and a prompt to press any key.

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char string[1000];
6     printf("Input a string to convert to lower case\n");
7     gets(string);
8     printf("Input string in lower case: %s\n",strlwr(string));
9     return 0;
10 }
```

D:\swaroop\ssss.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

SSSS.C

D:\swaroop\ssss.exe

Input a string to convert to lower case  
chandana chanikya  
Input string in lower case: chandana chanikya

-----

Process exited after 24.79 seconds with return value 0  
Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Line: 8 Col: 62 Sel: 0 Lines: 10 Length: 233 Insert Done parsing in 0.015 seconds

28°

Search

IN DEV

02:31 30-03-2023

21. Reverse a String in C: Reversing a string means replacing the last element in the first position and vice versa and doing it for all characters in a string. For example, the first character is swapped with the last character, the second character is swapped with the second last character and so on.

A string whose reverse string is the same as the original string is called a palindrome.

### **Examples:**

**Input string:** “hello”

**Reversed String:** “olleh”

**Input string:** “Sanfoundry C Programming”

**Reversed string:** “gnimmargorP C yrdnuofnaS”

The screenshot shows the Dev-C++ IDE interface. The top menu bar includes File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Build. The status bar at the bottom displays the current line (Line: 5), column (Col: 19), selection (Sel: 0), lines (Lines: 10), length (Length: 260), and the time taken for parsing (Done parsing in 0.016 seconds). The bottom right corner shows the date and time (30-03-2023, 02:44 AM).

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

SSSS.C

```
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char str[40];
6     printf ("\n Enter a string to be reversed: ");
7     scanf ("%s", str);
8     printf ("\n After the reverse of a string: %s ", strrev(str));
9     return 0;
10 }
```

D:\swaroop\ssss.exe

Enter a string to be reversed: ambulance

After the reverse of a string: ecnalubma

Process exited after 17.8 seconds with return value 0

Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Line: 5 Col: 19 Sel: 0 Lines: 10 Length: 260 Insert Done parsing in 0.016 seconds

28°

Search

28° ENG IN 02:44 30-03-2023

22. Write a C program to find length of a string using loop. How to find length of a string without using in-built library function `strlen()` in C programming. Effective way to find length of a string without using `strlen()` function. How to find length of a string using `strlen()` string function.

## Example

## Input

Input string: I love programming. I love Codeforwin.

# Output

Length of string: 38

The screenshot shows the Dev-C++ IDE interface. On the left, the code editor displays a file named 'ssss.c' containing the following C code:

```

1 #include <stdio.h>
2 int main()
3 {
4     char str[100];
5     int i,length=0;
6     printf("Enter a string: \n");
7     scanf("%s",str);
8     for(i=0; str[i]!='\0'; i++)
9     {
10         length++;
11     }
12     printf("\nLength of input string: %d",length);
13     return 0;
14 }

```

On the right, a terminal window titled 'D:\swaroop\ssss.exe' shows the program's output:

```

Enter a string:
prepinsta
Length of input string: 9
-----
Process exited after 13.89 seconds with return value 0
Press any key to continue . . .

```

The status bar at the bottom of the IDE provides compilation information: 'Done parsing in 0.016 seconds'. The taskbar at the very bottom shows various application icons.

23. Write a C program to remove all occurrences of a given character from the string using loop. Write a function to remove all occurrences of a character from a string.  
 How to remove all occurrences of a character from the string in C programming.  
 Logic to remove all occurrences of a character from a given string in C program.

### **Example**

#### **Input**

Input string : I Love Programming. I Love Codeforwin.

Input character to remove : 'I'

#### **Output**

String after removing all 'I' : Love Programming. Love Codeforwin.

```

D:\swaroop\ssss.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(ssss.c) (globals)
5555.C
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char s[1000],c,temp=1;
6     int i,j,k=0,n;
7     printf("Enter the string : ");
8     gets(s);
9     printf("Enter character: ");
10    c=getchar();
11    for(i=0;s[i];i++)
12    {
13        s[i]=s[i*k];
14        if(s[i]==c)
15        {
16            k++;
17            i--;
18        }
19    }
20    printf("%s",s);
21    return 0;
22 }

```

D:\swaroop\ssss.exe

Enter the string : welcome to cbeginers  
 Enter character: n  
 welcome to cbegiers  
 -----  
 Process exited after 39.02 seconds with return  
 value 0  
 Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Line: 19 Col: 6 Sel: 0 Lines: 22 Length: 357 Insert Done parsing in 0.015 seconds

28° Search

28° Search

24. Write a C program to remove extra spaces, blanks from a string. How to remove extra blank spaces, blanks from a given string using functions in C programming. Logic to remove extra white space characters from a string in C.

### Example

#### Input

Input string: Learn C programming at Codeforwin.

#### Output

String after removing extra blanks:

"Learn C programming at Codeforwin"

The screenshot shows the Dev-C++ IDE interface. The code editor window displays a file named 'ssss.c' containing a C program. The program includes headers for stdio.h and string.h, defines a function void deleteblankspaces(char \*s), and implements it using a loop to iterate through the string. The main() function reads a string from the user, calls deleteblankspaces(), and prints the result. The terminal window shows the execution of the program, where it prompts for a string ('Enter the string : a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z') and then prints the string after removing all duplicates ('string after removing all duplicates:a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z'). The status bar at the bottom provides system information like temperature (28°), search bar, taskbar icons, and system tray.

```

1 #include <stdio.h>
2 #include <string.h>
3 void deleteblankspaces(char *s)
4 {
5     int i,k=0;
6     for(i=0;s[i];i++)
7     {
8         s[i]=s[i+k];
9         if(s[i]==' '|| s[i]=='\t')
10        {
11            k++;
12            i--;
13        }
14    }
15 }
16 int main()
17 {
18     char s[1000];
19     printf("Enter the string : ");
20     gets(s);
21     deleteblankspaces(s);
22     printf("string after removing all duplicates:");
23     printf("%s",s);
24     return 0;
25 }

```

25. Write a C program to trim both leading and trailing white space characters in a string using loop. How to remove both leading and trailing white space characters in a string using loop in C programming. Logic to delete all leading and trailing white space characters from a given string in C.

### **Example**

#### **Input**

Input string: " Lots of leading and trailing spaces. "

#### **Output**

String after removing leading and trailing white spaces:  
"Lots of leading and trailing spaces."

D:\sварооп\ssss.c - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
5555.C
1 #include <stdio.h>
2 #include <string.h>
3 int main()
4 {
5     char s[1000];
6     int i,j;
7     printf("Enter any string : ");
8     gets(s);
9     printf("Before trimming leading and trailing white spaces :'%s'",s);
10    for(i=0;s[i]==' '||s[i]=='\t';i++);
11    for(j=0;s[i];i++)
12    {
13        s[j++]=s[i];
14    }
15    s[j]='\0';
16    for(i=0;s[i]!='\0';i++)
17    {
18        if(s[i]!=' ' && s[i]!='\t')
19            j=i;
20    }
21    s[j+1]='\0';
22    printf("\nAfter trimming leading and trailing white spaces :'%s'",s);
23 }
```

D:\sварооп\ssss.exe

Enter any string : removing leading and trailing white spaces

Before trimming leading and trailing white spaces :'removing leading and trailing white spaces'

After trimming leading and trailing white spaces :'removing leading and trailing white spaces'

-----

Process exited after 54.73 seconds with return value 95

Press any key to continue . . . |

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Line: 10 Col: 40 Sel: 0 Lines: 23 Length: 476 Insert Done parsing in 0.016 seconds

28° Search

ENG IN 03:09 30-03-2023