

Advanced C Lab assignment 2 Ex3

Name: Sankalp Mukim

Registration Number: 20BDS0128

Course Code: CSE2010

Slot: L27+L28

Table of Contents

Part 1	4
WAP to sort n alphabets (both upper and lower) in an array in ascending order.	4
Code	4
Output.....	5
WAP to extract odd, even, and prime numbers in an array to 3 different arrays and display this array with proper formatting	5
Code	5
Output.....	6
WAP to rearrange positive and negative integers in a 1-D array.....	6
Code	6
Output.....	8
WAP to insert an element into the array, considering all the 3 cases i.e., Beginning of array, Middle of Array, End of array. Finally display the resultant array.	8
Code	8
Output.....	9
WAP a program to swap adjacent elements in a one-dimensional array	9
Code	9
Output.....	10
WAP to search an element in an array and replace with an element *	10
Code	10
Output.....	11
WAP to split the given 1-D array into repeating and non-repeating elements of the array	11
Code	11
Output.....	12
WAP to find the sum of two matrix.....	13
Code	13
Output.....	14
WAP to transpose the given matrix.....	14

Code	14
Output.....	15
WAP to find the product of two matrix	15
Code	15
Output.....	16
WAP to find the product of boundary elements of a given matrix.	16
Algorithm	16
Code	17
Output.....	18
WAP to find the maximum element in the given row and minimum element in the given column for a given matrix	18
Algorithm	18
Code	18
Output.....	19
WAP to carry out Left to Right rotations of a 1-D array n times	19
Algorithm	19
Code	20
Output.....	20
WAP to find all the patterns of 0(2+)0 in the given string. Given a string containing 0's and 2's. Find the total number of 0(2+)0 patterns in the string and output it.	21
Algorithm	21
Code	21
Output.....	22
Part 2	22
Write a C program to count maximum and minimum occurrence for a given character ch in the given string s1.	22
Code	22
Output.....	23
WAP a program to reverse the given string and display the resultant string	23
Code	23
Output.....	23
Write a C program to concatenate the following quotes of Dr. A. P. J “If you fail, never give up because FAIL means “First Attempt in Learning.” “All of us do not have equal talent. But, all of us have an equal opportunity to develop our talents.” “Excellence is a continuous process & not an accident” Finally display the output in the following format Dr. A. P. J quotes follows.....(with the resultant string).....	24
Code	24

Output.....	24
Considering the given quotes of Dr. A. P. J stated above, find the frequency of occurrence of all the special characters in the given quotes.....	24
Code	24
Output.....	25
Write a C program to count the number of words, number of sentences and finally the length of the string without using any built- in functions of strings.	25
Code	25
Output.....	26
WAP to remove all characters from the given string except the alphabet character and display the final string	26
Code	26
Output.....	27
Write a program in C to replace the spaces of a given string “ARISE, AWAKE, and STOP NOT UNTIL THE GOAL is ACHIEVED” with a special character of your choice.	27
Code	27
Output.....	27
WAP to split the given sentence into words and display the words one after the other.	27
Code	27
Output.....	28
Write a program to find the number of times a given word 'the" appears in the given string “Best brain of the nation may be found in the last benches of the classroom”.	28
Algorithms.....	28
Code	29
Output.....	29
WAP to remove the duplicate characters from the given string, S and print the resultant string, RS (Hint: S <-AabCaBcdEGzYz , RS <-AbCdEGzY)	29
Code	29
Output.....	30
WAP to remove all characters from the given string except the alphabet character and display the final string (without using the second string)	30
Code	30
Output.....	31
WAP to insert a substring anywhere in between the given string except in the first and last position of the string. Finally display the resultant string on inserting the sub-string.....	31
Code	31
Output.....	32
WAP to reverse words in the given strings and display the reversed string	32

Code.....	32
Output.....	33

Part 1

WAP to sort n alphabets (both upper and lower) in an array in ascending order.

Code

```
#include <stdio.h>

int main()
{
    printf("Enter the number of characters:");
    int n;
    scanf("%d", &n);
    printf("Enter %d characters:\n", n);
    char arr[50];
    scanf("%c", &arr[0]);
    for (int i = 0; i < n; i++)
    {
        scanf("%c", &(arr[i]));
    }

    printf("Before sorting:\n[\n\t");
    for (int i = 0; i < n; i++)
    {
        printf("%c, ", arr[i]);
    }
    printf("\n]");
    int min_idx;
    // selection sort
    for (int i = 0; i < n - 1; i++)
    {
        min_idx = i;
        for (int j = i + 1; j < n; j++)
        {
            if (arr[j] < arr[min_idx])
            {
                min_idx = j;
            }
        }

        // swap
        char temp = arr[min_idx];
        arr[min_idx] = arr[i];
        arr[i] = temp;
    }
    printf("\nAfter sorting\n[\n\t");
```

```

    for (int i = 0; i < n; i++)
    {
        printf("%c, ", arr[i]);
    }
    printf("\n\n");
    return 0;
}

```

Output

```

sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> gcc -o q1 q1.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> ./q1
Enter the number of characters:15
Enter 15 characters:
bcjhkjgfvfbhhrt0
Before sorting:
[
    b, c, j, h, k, j, g, v, f, b, h, h, r, t, 0,
]
After sorting
[
    0, b, b, c, f, g, h, h, h, j, j, k, r, t, v,
]
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
>

```

WAP to extract odd, even, and prime numbers in an array to 3 different arrays and display this array with proper formatting

Code

```

#include <stdio.h>

int main()
{
    int arr[100], odds[100], evens[100], primes[100];
    int n;
    printf("Enter n: ");
    scanf("%d", &n);
    printf("Enter arr: ");
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    printf("\n");
    int odd_count = 0, even_count = 0, prime_count = 0;
    for (int i = 0; i < n; i++)
    {
        if (arr[i] % 2 == 0)
        {
            evens[even_count++] = arr[i];
        }
        else
        {
            odds[odd_count++] = arr[i];
        }
    }
}

```

```

    }
    int flag = 0;
    for (int j = 2; j <= arr[i] / 2; j++)
    {
        if (arr[i] % j == 0)
        {
            flag = 1;
            break;
        }
    }
    if (flag == 0)
    {
        primes[prime_count++] = arr[i];
    }
}
printf("Odds: ");
for (int i = 0; i < odd_count; i++)
{
    printf("%d, ", odds[i]);
}
printf("\nEvens: ");
for (int i = 0; i < even_count; i++)
{
    printf("%d, ", evens[i]);
}
printf("\nPrimes: ");
for (int i = 0; i < prime_count; i++)
{
    printf("%d, ", primes[i]);
}
printf("\n");
return 0;
}

```

Output

```

sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 |main
> gcc -o q2 q2.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 |main
> ./q2
Enter n: 5
Enter arr: 2 23 3 4 45

Odds: 23, 3, 45,
Evens: 2, 4,
Primes: 2, 23, 3,
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 |main
>

```

WAP to rearrange positive and negative integers in a 1-D array

Code

```
#include <stdio.h>
```

```

// Arranging negative then positive values
int main()
{
    int n, arr[50];
    printf("Enter n:");
    scanf("%d", &n);
    printf("Enter %d numbers:\n", n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    printf("\n");
    // traverse the array
    for (int i = 0; i < n; i++)
    {
        // if the element is negative
        if (arr[i] > 0)
        {
            // traverse the array again
            for (int j = i + 1; j < n; j++)
            {
                // if the element is positive
                if (arr[j] < 0)
                {
                    // swap
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                    break;
                }
            }
        }
    }
    printf("\nAfter sorting:\n[\n\t");
    for (int i = 0; i < n; i++)
    {
        printf("%d, ", arr[i]);
    }
    printf("\n]\n");
    return 0;
}

```

Output

```
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 |main
> gcc -o q3 q3.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 |main
> ./q3
Enter n:10
Enter 10 numbers:
1 -1 -3 5 6 7 -2 -5 -22 5

After sorting:
[
    -1, -3, -2, -5, -22, 7, 1, 5, 6, 5,
]
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 |main
>
```

WAP to insert an element into the array, considering all the 3 cases i.e., Beginning of array, Middle of Array, End of array. Finally display the resultant array.

Code

```
#include <stdio.h>

int main()
{
    int arr[100] = {10, 20, 30};
    int SIZE = 3;
    int i, j;
    printf("Array to start with:\n[\n\t");
    for (i = 0; i < SIZE; i++)
    {
        printf("%d, ", arr[i]);
    }
    printf("\n]\n");
    printf("Enter number to add to beginning of array: ");
    int num;
    scanf("%d", &num);
    for (i = SIZE; i > 0; i--)
    {
        arr[i] = arr[i - 1];
    }
    arr[0] = num;
    SIZE++;
    printf("After adding %d to beginning of array:\n[\n\t", num);
    for (i = 0; i < SIZE; i++)
    {
        printf("%d, ", arr[i]);
    }
    printf("\n]\n");
    printf("Enter element index to add to array: ");
    int index, element;
    scanf("%d", &index);
    printf("Enter element to add to array: ");
```



```

scanf("%d", &element);
for (int i = SIZE; i > index; i--)
{
    arr[i] = arr[i - 1];
}
arr[index] = element;
SIZE++;
printf("After adding %d to middle of array:\n[\n\t", element);
for (i = 0; i < SIZE; i++)
{
    printf("%d, ", arr[i]);
}
printf("\n]\n");
printf("Enter element to add to end of array: ");
scanf("%d", &arr[SIZE++]);
printf("After adding element to end of array:\n[\n\t");
for (i = 0; i < SIZE; i++)
{
    printf("%d, ", arr[i]);
}
printf("\n]\n");
}

```

Output

```

> sankaa@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> gcc -o q4 q4.c
> sankaa@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> ./q4
Array to start with:
[
    10, 20, 30,
]
Enter number to add to beginning of array: 5
After adding 5 to beginning of array:
[
    5, 10, 20, 30,
]
Enter element index to add to array: 2
Enter element to add to array: 25
After adding 25 to middle of array:
[
    5, 10, 25, 20, 30,
]
Enter element to add to end of array: 45
After adding element to end of array:
[
    5, 10, 25, 20, 30, 45,
]
> sankaa@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
>

```

WAP a program to swap adjacent elements in a one-dimensional array

Code

```

#include <stdio.h>

int main()
{
    int n, arr[100];
    printf("Enter n: ");
}

```

```

scanf("%d", &n);
for (int i = 0; i < n; i++)
{
    scanf("%d", &arr[i]);
}
for (int i = 1; i < n; i += 2)
{
    int temp = arr[i];
    arr[i] = arr[i - 1];
    arr[i - 1] = temp;
}
printf("Array after swapping:\n");
printf("[\n\t");
for (int i = 0; i < n; i++)
{
    printf("%d, ", arr[i]);
}
printf("\n]\n");
}

```

Output

```

sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> gcc -o q5 q5.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> ./q5
Enter n: 5
1 2 3 4 5
Array after swapping:
[
    2, 1, 4, 3, 5,
]
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
>

```

WAP to search an element in an array and replace with an element *

Code

```

#include <stdio.h>

int main()
{
    printf("Enter the number of characters:");
    int n;
    scanf("%d", &n);
    printf("Enter %d characters:\n", n);
    char arr[50];
    scanf("%c", &arr[0]);
    for (int i = 0; i < n; i++)
    {
        scanf("%c", &(arr[i]));
    }

    printf("Inputted array:\n[\n\t");
    for (int i = 0; i < n; i++)

```

```

{
    printf("%c, ", arr[i]);
}
printf("\n\n");

printf("Enter the element to search and replace: ");
char ch;
scanf("%c", &ch);
scanf("%c", &ch);
for (int i = 0; i < n; i++)
{
    if (arr[i] == ch)
    {
        arr[i] = '*';
    }
}
printf("Output array:\n[\n\t");
for (int i = 0; i < n; i++)
{
    printf("%c, ", arr[i]);
}
printf("\n\n");
return 0;
}

```

Output

```

sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 |main
> gcc -o q6 q6.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 |main
> ./q6
Enter the number of characters:5
Enter 5 characters:
abcde
Inputted array:
[
    a, b, c, d, e,
]
Enter the element to search and replace: c
Output array:
[
    a, b, *, d, e,
]
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 |main
>

```

WAP to split the given 1-D array into repeating and non-repeating elements of the array

Code

```

#include <stdio.h>

int main()
{
    int n;
    printf("Enter n: \n");
    scanf("%d", &n);
}

```

```

printf("Enter array: \n");
int arr[50];
for (int i = 0; i < n; i++)
{
    scanf("%d", &arr[i]);
}
printf("\n");
int last_non_repeating_element = -1;
for (int i = 0; i < n; i++)
{
    int repeating = 0;
    for (int j = 0; j < n; j++)
    {
        if (arr[i] == arr[j] && i != j)
        {
            repeating++;
            break;
        }
    }
    if (!repeating)
    {
        int non_repeating = arr[i];
        for (int k = i; k > last_non_repeating_element; k--)
        {
            arr[k] = arr[k - 1];
        }
        arr[++last_non_repeating_element] = non_repeating;
    }
}
printf("Non repeating to the left and repeating to the right\n");
printf("Final array is: \n\n\t");
for (int i = 0; i < n; i++)
{
    printf("%d, ", arr[i]);
}
printf("\n\n");
}

```

Output

```

sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 main
> gcc -o q7 q7.c
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 main
> ./q7
Enter n:
10
Enter array:
1 2 5 2 3 4 7 4 2 4

Non repeating to the left and repeating to the right
Final array is:
[
    1, 5, 3, 7, 2, 2, 4, 4, 2, 4,
]
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 main
>

```

WAP to find the sum of two matrix

Code

```
#include <stdio.h>

int main()
{
    int m, n;
    printf("Enter no. of rows(m) and no. of columns(n).\n");
    scanf("%d", &m);
    scanf("%d", &n);
    int mat1[20][20], mat2[20][20], mat3[20][20];
    printf("Enter matrix 1 rowwise\n");
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%d", &mat1[i][j]);
        }
    }
    printf("Enter matrix 2 rowwise\n");
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%d", &mat2[i][j]);
            mat3[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
    printf("Sum of matrix 1 & 2 is:\n");
    printf("[");
    for (int i = 0; i < m; i++)
    {
        printf("\n\t");
        for (int j = 0; j < n; j++)
        {
            printf("%d  ", mat3[i][j]);
        }
    }
    printf("\n]\n");
}
```

Output

```
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> gcc -o q7 q7.c
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> ./q7
Enter n:
10
Enter array:
1 2 5 2 3 4 7 4 2 4

Non repeating to the left and repeating to the right
Final array is:
[
    1, 5, 3, 7, 2, 2, 4, 4, 2, 4,
]
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
>
```

WAP to transpose the given matrix

Code

```
#include <stdio.h>

int main()
{
    int m, n;
    printf("Enter no. of rows(m) and no. of columns(n).\n");
    scanf("%d", &m);
    scanf("%d", &n);
    int mat1[20][20], mat2[20][20];
    printf("Enter matrix 1 rowwise\n");
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%d", &mat1[i][j]);
        }
    }
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            mat2[i][j] = mat1[j][i];
        }
    }
    printf("Transpose of given matrix:\n");
    printf("[");
    for (int i = 0; i < n; i++)
    {
        printf("\n\t");
        for (int j = 0; j < m; j++)
        {
            printf("%d  ", mat2[i][j]);
        }
    }
    printf("\n]\n");
}
```

```
}
```

Output

```
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> gcc -o q9 q9.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> ./q9
Enter no. of rows(m) and no. of columns(n).
5 5
Enter matrix 1 rowwise
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
Transpose of given matrix:
[
    1  6 11 16 21
    2  7 12 17 22
    3  8 13 18 23
    4  9 14 19 24
    5 10 15 20 25
]
```

WAP to find the product of two matrix

Code

```
#include <stdio.h>

int main()
{
    int m1, n1, m2, n2;
    int mat1[20][20], mat2[20][20], mat3[20][20];
    printf("Enter no. of rows(m) and no. of columns(n) for 1st matrix.\n");
    scanf("%d", &m1);
    scanf("%d", &n1);
    printf("Enter no. of rows(m) and no. of columns(n) for 2nd matrix.\n");
    scanf("%d", &m2);
    scanf("%d", &n2);
    if (n1 != m2)
    {
        printf("Matrix multiplication not possible.\n");
        return 0;
    }
    printf("Enter matrix 1 rowwise\n");
    for (int i = 0; i < m1; i++)
    {
        for (int j = 0; j < n1; j++)
        {
            scanf("%d", &mat1[i][j]);
        }
    }
    printf("Enter matrix 2 rowwise\n");
    for (int i = 0; i < m2; i++)
```

```

{
    for (int j = 0; j < n2; j++)
    {
        scanf("%d", &mat2[i][j]);
    }
}
for (int i = 0; i < m1; i++)
{
    for (int j = 0; j < n2; j++)
    {
        mat3[i][j] = 0;
        for (int k = 0; k < n1; k++)
        {
            mat3[i][j] += mat1[i][k] * mat2[k][j];
        }
    }
}
printf("Product of matrix 1 & 2 is:\n");
printf("[");
for (int i = 0; i < m1; i++)
{
    printf("\n\t");
    for (int j = 0; j < n2; j++)
    {
        printf("%d  ", mat3[i][j]);
    }
}
printf("\n]\n");
}

```

Output

```

sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> gcc -o q10 q10.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> ./q10
Enter no. of rows(m) and no. of columns(n) for 1st matrix.
2 3
Enter no. of rows(m) and no. of columns(n) for 2nd matrix.
3 2
Enter matrix 1 rowwise
1 2 3
4 5 6
Enter matrix 2 rowwise
1 2
3 4
5 6
Product of matrix 1 & 2 is:
[
    22 28
    49 64
]
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
>

```

WAP to find the product of boundary elements of a given matrix.

Algorithm

1. Take the matrix from the user.

2. Declare a variable to store the product.
3. Traverse the matrix.
4. If the element is the first element of the row, then multiply it with the product.
5. Else if the element is the first element of the column, then multiply it with the product.
6. Else if the element is the last element of the row, then multiply it with the product.
7. Else if the element is the last element of the column, then multiply it with the product.
8. Display the product.

Code

```
#include <stdio.h>

int main()
{
    int m, n, product = 1;
    printf("Enter no. of rows(m) and no. of columns(n).\n");
    scanf("%d", &m);
    scanf("%d", &n);
    int mat[20][20];
    printf("Enter matrix 1 rowwise\n");
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%d", &mat[i][j]);
        }
    }
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (i == 0 || j == 0 || i == m - 1 || j == n - 1)
            {
                product *= mat[i][j];
            }
        }
    }
    printf("Product of boundary elements: %d\n", product);
}
```

Output

```
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> gcc -o q11 q11.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> ./q11
Enter no. of rows(m) and no. of columns(n).
4 4
Enter matrix 1 rowwise
1 1 1 1
1 0 0 1
1 0 0 1
1 3 4 1
Product of boundary elements: 12
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
```

WAP to find the maximum element in the given row and minimum element in the given column for a given matrix

Algorithm

1. Take the matrix from the user.
2. Take the row number from the user.
3. Take the column number from the user.
4. Declare a variable to store the maximum element.
5. Declare a variable to store the minimum element.
6. Traverse the row.
7. If the element is greater than the maximum element, then set the maximum element to the element.
8. Else if the element is less than the minimum element, then set the minimum element to the element.
9. Display the maximum element and the minimum element.

Code

```
#include <stdio.h>

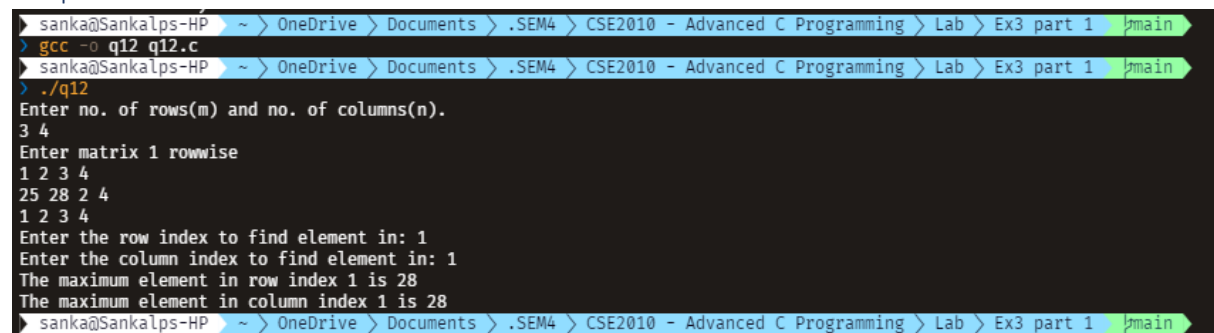
int main()
{
    int m, n;
    printf("Enter no. of rows(m) and no. of columns(n).\n");
    scanf("%d", &m);
    scanf("%d", &n);
    int mat[20][20];
    printf("Enter matrix 1 rowwise\n");
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%d", &mat[i][j]);
        }
    }
    int row, col, max;
```

```

printf("Enter the row index to find element in: ");
scanf("%d", &row);
printf("Enter the column index to find element in: ");
scanf("%d", &col);
// row
max = mat[row][0];
for (int i = 0; i < n; i++)
{
    if (mat[row][i] > max)
    {
        max = mat[row][i];
    }
}
printf("The maximum element in row index %d is %d\n", row, max);
// column
max = mat[0][col];
for (int i = 0; i < m; i++)
{
    if (mat[i][col] > max)
    {
        max = mat[i][col];
    }
}
printf("The maximum element in column index %d is %d\n", col,
max);
}

```

Output



```

sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> gcc -o q12 q12.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> ./q12
Enter no. of rows(m) and no. of columns(n).
3 4
Enter matrix 1 rowwise
1 2 3 4
25 28 2 4
1 2 3 4
Enter the row index to find element in: 1
Enter the column index to find element in: 1
The maximum element in row index 1 is 28
The maximum element in column index 1 is 28
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main

```

WAP to carry out Left to Right rotations of a 1-D array n times

Algorithm

1. Take the array from the user.
2. Take the number of rotations from the user.
3. For n rotations,
4. Traverse the array and shift the elements to the right.
5. Set the last element to 0.

6. Display the array.

Code

```
#include <stdio.h>

int main()
{
    int n, SIZE;
    printf("Enter size of array: ");
    scanf("%d", &SIZE);
    printf("Enter %d elements: ", SIZE);
    int arr[50];
    for (int i = 0; i < SIZE; i++)
    {
        scanf("%d", &arr[i]);
    }
    printf("\n");
    printf("Enter the number of rotations: ");
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
    {
        int last = arr[SIZE - 1];
        for (int j = SIZE - 1; j > 0; j--)
        {
            arr[j] = arr[j - 1];
        }
        arr[0] = last;
    }
    printf("\n");
    printf("Array after %d left to right rotations:\n[\n\t", n);
    for (int i = 0; i < SIZE; i++)
    {
        printf("%d, ", arr[i]);
    }
    printf("\n]\n");
}
```

Output

```
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> gcc -o q13 q13.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> ./q13
Enter size of array: 5
Enter 5 elements: 1 2 3 4 5

Enter the number of rotations: 3

Array after 3 left to right rotations:
[
    3, 4, 5, 1, 2,
]
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
>
```

WAP to find all the patterns of 0(2+)0 in the given string. Given a string containing 0's and 2's. Find the total number of 0(2+)0 patterns in the string and output it.

Algorithm

1. Take string from the user.
2. Declare a variable to store the count of 0(2+)0 patterns.
3. Traverse the string.
4. If the character is 0, then continue.
5. Else if the character is 2, and if the last character is a 0, then
6. Keep traversing the string and check if the next character is 0 or 2.
7. If the next character is 0, then increment the count and set last_zero to 1.
8. Else if the next character is 2, then continue.
9. Display the count.

Code

```
#include <stdio.h>

int main()
{
    char string[1024];
    printf("Enter a string of 0's and 2's: ");
    scanf("%s", string);
    int i = 0, last_zero = 0, count = 0;
    while (string[i])
    {
        if (string[i] == '0')
        {
            last_zero = 1;
            i++;
            continue;
        }
        else if (string[i] == '2' && last_zero)
        {
            last_zero = 0;
            while (string[i] && string[i] == '2')
            {
                i++;
            }
            // making sure loop exited on 0 instead of null
            if (string[i] == '0')
            {
                count++;
                last_zero = 1;
            }
        }
    }
}
```

```

        i++;
    }
    printf("Number of patterns: %d\n", count);
}

```

Output

```

sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> gcc -o q14 q14.c
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
> ./q14
Enter a string of 0's and 2's: 2202202222020
Number of patterns: 3
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 1 > main
>

```

Part 2

Write a C program to count maximum and minimum occurrence for a given character ch in the given string s1.

Code

```

#include <stdio.h>

int main()
{
    char string[200], max_char = '\0', min_char = '\0';
    int max_count = 0, min_count = 201;
    printf("Enter string: ");
    scanf("%[^\n]s", string);
    getchar();
    for (int i = 0; string[i]; i++)
    {
        if (max_char != string[i] || min_count != string[i])
        {
            int count = 0;
            for (int j = 0; string[j]; j++)
            {
                if (string[j] == string[i])
                {
                    count++;
                }
            }
            if (count > max_count)
            {
                max_count = count;
                max_char = string[i];
            }
            if (count < min_count)
            {
                min_count = count;
                min_char = string[i];
            }
        }
    }
}

```

```

    }
}
printf("The character that appeared the maximum times is: %c\n",
max_char);
printf("The character that appeared the minimum times is: %c\n",
min_char);
}

```

Output

```

sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
> gcc -o q1 q1.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
> ./q1
Enter string: aaaabbbcccdde
The character that appeared the maximum times is: a
The character that appeared the minimum times is: e
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
>

```

WAP a program to reverse the given string and display the resultant string

Code

```

#include <stdio.h>

int main()
{
    char string[200], reversed[200];
    printf("Enter string to reverse: ");
    scanf("%[^\n]s", string);
    getchar();
    int length;
    for (length = 0; string[length]; length++)
        ;
    for (int i = length - 1, j = 0; i >= 0; i--, j++)
    {
        reversed[j] = string[i];
    }
    reversed[length] = '\0';
    printf("Reversed string: %s\n", reversed);
}

```

Output

```

sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
> gcc -o q2 q2.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
> ./q2
Enter string to reverse: abcdefghijkl
Reversed string: lkjihgfedcba
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
>

```

Write a C program to concatenate the following quotes of Dr. A. P. J

"If you fail, never give up because FAIL means "First Attempt in Learning."

"All of us do not have equal talent. But, all of us have an equal opportunity to develop our talents."

"Excellence is a continuous process & not an accident"

Finally display the output in the following format

Dr. A. P. J quotes follows.....(with the resultant string).

Code

```
#include <stdio.h>
#include <string.h>

int main()
{
    char main_quote[] = "Dr. A. P. J. quotes follows....\n";
    char quote1[] = "If you fail, never give up because FAIL means\n\"First Atetmpt in Learning.\n\"";
    char quote2[] = "All of us do not have equal talent. But, all of us have an equal opportunity to develop our talents.";
    char quote3[] = "Excellence is a continuous process & not an accident";
    strcat(main_quote, quote1);
    strcat(main_quote, quote2);
    strcat(main_quote, quote3);
    printf("%s\n", main_quote);
}
```

Output

```
> sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
> gcc -o q3 q3.c
> sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
> ./q3
Dr. A. P. J. quotes follows....
If you fail, never give up because FAIL means "First Atetmpt in Learning."All of us do not have equal talent. But, al
ident
```

Considering the given quotes of Dr. A. P. J stated above, find the frequency of occurrence of all the special characters in the given quotes.

Code

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main()
{
    char main_quote[] = "Dr. A. P. J. quotes follows....\n";
    char quote1[] = "If you fail, never give up because FAIL means\n\"First Atetmpt in Learning.\n\"";
```



```

    char quote2[] = "All of us do not have equal talent. But, all of
us have an equal opportunity to develop our talents.";
    char quote3[] = "Excellence is a continuous process & not an
accident";
    strcat(main_quote, quote1);
    strcat(main_quote, quote2);
    strcat(main_quote, quote3);
    int count = 0;
    for (int i = 0; main_quote[i]; i++)
        if (ispunct(main_quote[i]))
            count++;
    printf("Total number of special characters in APJ's quotes:
%d\n", count);
}

```

Output

```

sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
> gcc -o q4 q4.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
> ./q4
Total number of special characters in APJ's quotes: 9
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
>

```

Write a C program to count the number of words, number of sentences and finally the length of the string without using any built-in functions of strings.

Code

```

#include <stdio.h>

int main()
{
    int words = 0, sentences = 0, length;
    char string[200];
    printf("Enter a string: ");
    scanf("%[^\n]s", string);
    getchar();
    // length
    for (length = 0; string[length]; length++)
        ;
    // words and sentences
    int i = 0, sentence_start = 0, word_start = 0;
    while (string[i])
    {
        if (string[i] == ' ')
        {
            word_start = i + 1;
            words++;
        }
        if (string[i] == '.' || string[i] == '!' || string[i] == '?')
        {
            sentences++;
        }
        i++;
    }
}

```

```

        sentences++;
        sentence_start = i + 1;
    }
    i++;
}
printf("Words: %d\nSentences: %d\nLength: %d\n", words,
sentences, length);
}

```

Output

```

sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> gcc -o q5 q5.c
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> ./q5
Enter a string: If you fail, never give up because FAIL means "First Attempt in Learning."
Words: 12
Sentences: 1
Length: 74
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
>

```

WAP to remove all characters from the given string except the alphabet character and display the final string

Code

```

#include <stdio.h>

int main()
{
    char string[200];
    printf("Enter a string: ");
    scanf("%[^\\n]s", string);
    getchar();
    int i = 0, j = 0;
    while (string[i])
    {
        if (string[i] >= 'a' && string[i] <= 'z' || string[i] >= 'A'
&& string[i] <= 'Z')
        {
            string[j] = string[i];
            j++;
        }
        i++;
    }
    string[j] = '\\0';
    printf("%s\\n", string);
}

```

Output

```
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> gcc -o q6 q6.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> ./q6
Enter a string: abcd. acb. asjncd. kljbd, khjb ds.
abcdacb.asjncd.kljbdskhjbds
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
```

Write a program in C to replace the spaces of a given string “ARISE, AWAKE, and STOP NOT UNTIL THE GOAL is ACHIEVED” with a special character of your choice.

Code

```
#include <stdio.h>

int main()
{
    char string[] = "ARISE, AWAKE, and STOP NOT UNTIL THE GOAL is ACHIEVED";
    int i = 0, j = 0;
    while (string[i])
    {
        if (string[i] == ' ')
        {
            string[j] = '$';
            j++;
        }
        else
        {
            string[j] = string[i];
            j++;
        }
        i++;
    }
    string[j] = '\0';
    printf("%s\n", string);
}
```

Output

```
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> gcc -o q7 q7.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> ./q7
ARISE,$AWAKE,$and$STOP$NOT$UNTIL$THE$GOAL$is$ACHIEVED
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
```

WAP to split the given sentence into words and display the words one after the other.

Code

```
#include <stdio.h>
#include <stdlib.h>
```

```

int main()
{
    char *string = (char *)malloc(200);
    printf("Enter a string: ");
    scanf("%[^\n]s", string);
    getchar();
    int i = 0, j = 0;
    while (string[i])
    {
        if (string[i] == ' ')
        {
            string[i] = '\0';
            printf("%s\n", string);
            string = string + i + 1;
            i = 0;
        }
        i++;
    }
    string[i] = '\0';
    printf("%s\n", string);
}

```

Output

```

sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> gcc -o q8 q8.c
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> ./q8
Enter a string: WAP to split the given sentence into words and display the words one after the other.
WAP
to
split
the
given
sentence
into
words
and
display
the
words
one
after
the
other.
sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
>

```

Write a program to find the number of times a given word "the" appears in the given string "Best brain of the nation may be found in the last benches of the classroom".

Algorithms

1. Define the string.
2. Traverse the string.
3. If a character is a space, then check if the word "the" is present or not.
4. If the word "the" is present, then increment the count.
5. Else skip the character.

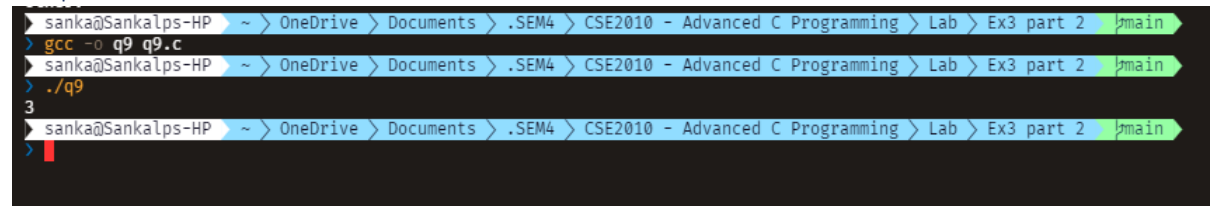
6. Display the count.

Code

```
#include <stdio.h>

int main()
{
    char string[] = "Best brain of the nation may be found in the last benches of the classroom";
    int i = 0, j = 0, count = 0;
    while (string[i])
    {
        if (string[i] == ' ')
        {
            j = i + 1;
            while (string[j] == ' ')
                j++;
            if (string[j] == 't' && string[j + 1] == 'h' && string[j + 2] == 'e')
                count++;
        }
        i++;
    }
    printf("%d\n", count);
}
```

Output



```
sanka@Sankalps-HP ~ > gcc -o q9 q9.c
sanka@Sankalps-HP ~ > ./q9
3
```

WAP to remove the duplicate characters from the given string, S and print the resultant string, RS (Hint: S <-AabCaBcdEGzYz , RS <-AbCdEGzY)

Code

```
#include <stdio.h>

char toLower(char c)
{
    if (c >= 'A' && c <= 'Z')
        return c + 32;
    else
        return c;
}

int charInArray(char c, char *array, int size)
```

```

{
    int i;
    for (i = 0; i < size; i++)
    {
        if (toLower(c) == toLower(array[i]))
            return 1;
    }
    return 0;
}

int main()
{
    char string[200];
    printf("Enter a string: ");
    scanf("%[^\n]s", string);
    getchar();
    char result_string[200];
    int length = 0;
    int i = 0;
    while (string[i])
    {
        if (charInArray(string[i], result_string, length))
        {
            i++;
            continue;
        }
        result_string[length] = string[i];
        length++;
        i++;
    }
    result_string[length] = '\0';
    printf("%s\n", result_string);
}

```

Output

```

sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> gcc -o q10 q10.c
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> ./q10
Enter a string: AabCaBcdEGzYz
AbCdEGzY
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
>

```

WAP to remove all characters from the given string except the alphabet character and display the final string (without using the second string)

Code

```

#include <stdio.h>

int main()
{

```

```

char string[200];
printf("Enter a string: ");
scanf("%[^\n]s", string);
getchar();
int i = 0, j = 0;
while (string[i])
{
    if (string[i] >= 'a' && string[i] <= 'z' || string[i] >= 'A'
&& string[i] <= 'Z')
    {
        string[j] = string[i];
        j++;
    }
    i++;
}
string[j] = '\0';
printf("%s\n", string);
}

```

Output

```

sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
> gcc -o q11 q11.c
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
> ./q11
Enter a string: ARISE, AWAKE, and STOP NOT UNTIL THE GOAL is ACHIEVED
ARISEAWAKEandSTOPNOTUNTILTHEGOALisACHIEVED
sanka@sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 |main
>

```

WAP to insert a substring anywhere in between the given string except in the first and last position of the string. Finally display the resultant string on inserting the substring.

Code

```

#include <stdio.h>
#include <stdlib.h>

// function to determine length of string
int strlen(char *string)
{
    int i = 0;
    while (string[i])
        i++;
    return i;
}

int main()
{
    char *string = (char *)malloc(200);
    printf("Enter a string: ");
    scanf("%[^\n]s", string);
    getchar();
}

```

```

char sub_string[200];
printf("Enter a sub-string: ");
scanf("%[^\\n]s", sub_string);
getchar();
printf("Enter the position: ");
int position, k = 0;
scanf("%d", &position);
while (string[position] != '\\0' && sub_string[k] != '\\0')
{
    int i;
    for (i = strlen(string); i >= position; i--)
    {
        string[i + 1] = string[i];
    }
    string[position] = sub_string[k];
    position++;
    k++;
}
printf("Output: %s\\n", string);
}

```

Output

```

> sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> gcc -o q12 q12.c
> sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> ./q12
Enter a string: abcde fghij
Enter a sub-string: klmno
Enter the position: 2
Output: abklmncde fghij
> sanka@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
>

```

WAP to reverse words in the given strings and display the reversed string

Code

```

#include <stdio.h>

void reverse(char *begin, char *end)
{
    char temp;
    while (begin < end)
    {
        temp = *begin;
        *begin++ = *end;
        *end-- = temp;
    }
}

// Function to reverse words*/
void reverseWords(char *s)
{
    char *word_begin = s;

```



```

// Word boundary
char *temp = s;

while (*temp)
{
    temp++;
    if (*temp == '\\0')
    {
        reverse(word_begin, temp - 1);
    }
    else if (*temp == ' ')
    {
        reverse(word_begin, temp - 1);
        word_begin = temp + 1;
    }
}

// Reverse the entire string
reverse(s, temp - 1);
}

// Driver Code
int main()
{
    char s[200];
    printf("Enter a string: ");
    scanf("%[^\\n]s", s);
    getchar();
    char *temp = s;
    reverseWords(s);
    printf("%s", s);
    return 0;
}

```

Output

```

> sankalps@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> gcc -o q13 q13.c
> sankalps@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
> ./q13
Enter a string: abcde fghij klmno pqrst uvwxy
uvwxy pqrst klmno fghij abcde
> sankalps@Sankalps-HP ~ > OneDrive > Documents > .SEM4 > CSE2010 - Advanced C Programming > Lab > Ex3 part 2 > main
>

```