DELHI TECHNOLOGICAL UNIVERSITY

# (FORMELY DELHI COLLEGE OF ENGINEERING)

**Department Of Software Engineering**



**DATA STRUCTURES**

**(SE-203)**

SUBMITTED TO: SUBMITTED BY:

Mr. Ankur Narwal Arjun Yadav 23/SE/32

Department Of Software Engineering

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Objective** | **Date** | **Sign** |
| **1.** | **To reverse an array of characters** | **21/08/24** |  |
| **2.** | **To perform various operations on array:**  **Insert element**  **Delete element**  **Find largest element**  **Find smallest element** | **04/09/24** |  |
| **3.** | **To perform string operations:**  **Merge 2 strings**  **Reverse a string**  **Find and replace substring** | **09/10/24** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**EXPERIMENT-1**

**AIM:** To take input from user and reverse an array.

**CODE:**

#include<stdio.h>

int main()

{

    int n, arr[n], i;

    printf("Enter the size of the array: ");

    scanf("%d", &n);

    printf("Enter the elements: ");

    for(i = 0; i < n; i++)

    {

        scanf("%d", &arr[i]);

    }

    int rev[n], j = 0;

    for(i = n-1; i >= 0; i--)

    {

        rev[j] = arr[i];

        j++;

    }

    printf("The Reversed array: ");

    for(i = 0; i < n; i++)

    {

        printf("%d ", rev[i]);

    }

}

**OUTPUT:**

**EXPERIMENT-2**

**AIM:** To perform various operations on array:

* Insert element
* Delete element
* Find largest element
* Find smallest element

**CODE:**

#include<stdio.h>

#define MAX\_size 100

int size = 0;

void inputarr(int arr[]);

void deletearr(int arr[]);

void largest(int arr[]);

void smallest(int arr[]);

void display(int arr[]);

void inputarr(int arr[]) {

    int n, elem;

    printf("Enter the number of elements you want to add: ");

    scanf("%d", &n);

    if (size + n > MAX\_size) {

        printf("Cannot add more elements. Array limit reached.\n");

        return;

    }

    for (int i = 0; i < n; i++) {

        printf("Enter element %d: ", size + 1);

        scanf("%d", &elem);

        arr[size++] = elem;

    }

}

void deletearr(int arr[]){

    if (size == 0) {

        printf("Array is empty. Nothing to delete.\n");

        return;

    }

    int num;

    printf("Enter the element to delete: ");

    scanf("%d", &num);

    for (int i = 0; i < size - 1; i++) {

        if (arr[i]==num){

            arr[i] = arr[i + 1];}

    }

    size--;

    printf("Element deleted.\n");

}

void largest(int arr[]){

    int largest=-1e7;

    for(int i=0;i<size;i++){

        if(largest<arr[i]){

            largest=arr[i];

        }

    }

    printf("largest element is %d \n",largest);

}

void display(int arr[]){

    for(int i=0;i<size;i++){

        printf("%d",arr[i]);

    }

}

void smallest(int arr[]){

    int smallest=1e7;

    for(int i=0;i<size;i++){

        if(smallest>arr[i]){

            smallest=arr[i];

        }

    }

    printf("largest element is %d \n",smallest);

}

int main(){

    int arr[MAX\_size];

    int flag=1;

    while(flag){

        printf("Please Select an input \n");

        printf("1. Enter elements in a one dimensional array \n");

        printf("2. delete element in a one dimensional array (have all conditions, beginning, last, middle index)  \n");

        printf("3, Find the largest element \n");

        printf("4. Find the smallest element \n");

        printf("5. diaplay\n");

        printf("6.exit \n");

        int n;

        scanf("%d",&n);

        switch(n){

            case 1:

            inputarr(arr);

            break;

            case 2:

            deletearr(arr);

            break;

            case 3:

            largest(arr);

            break;

            case 4:

            smallest(arr);

            break;

            case 5:

            display(arr);

            case 6:

            flag=0;

            break;

            default:

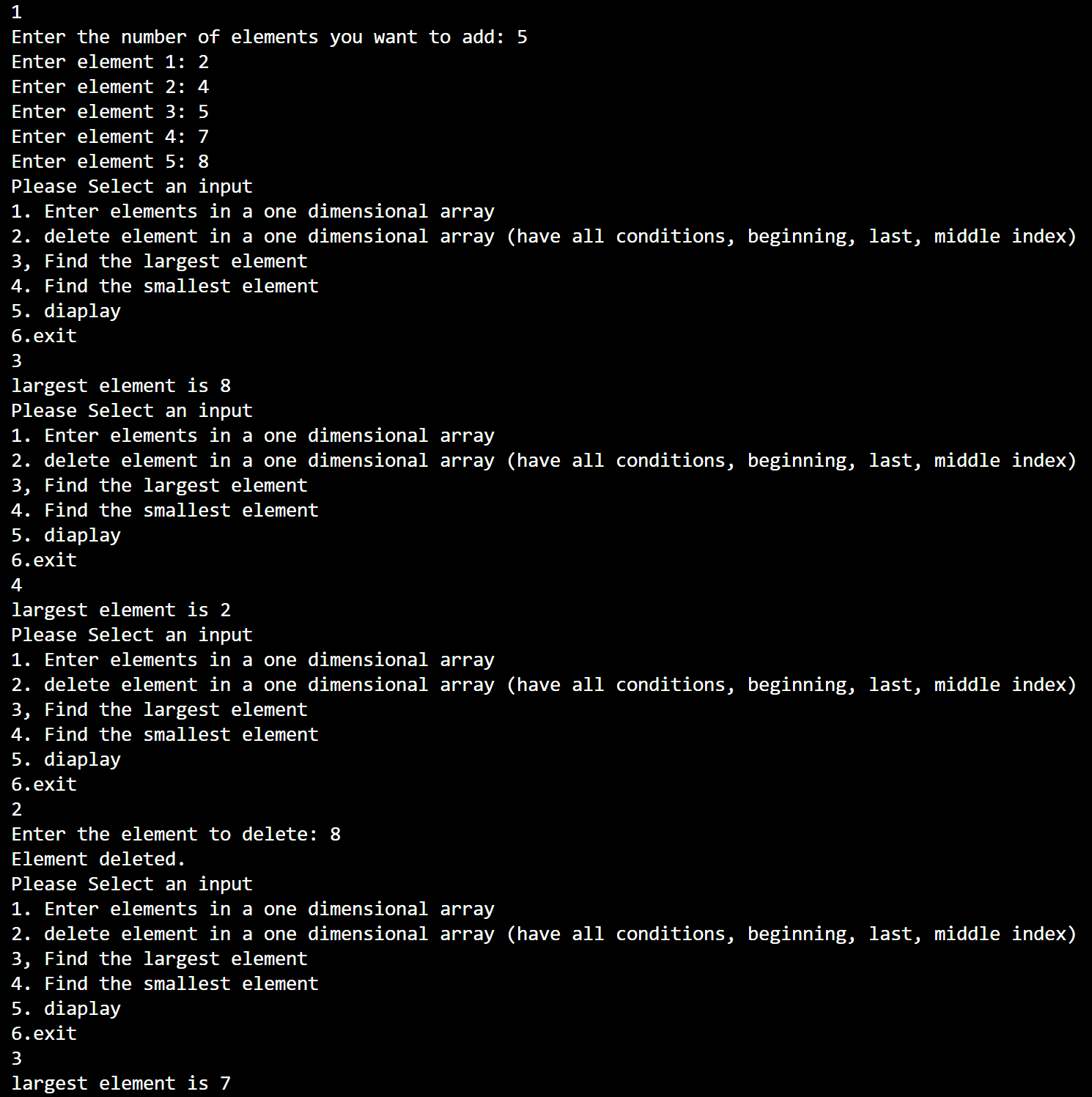
            printf("invalid input");

        }

    }

}

**OUTPUT:**

****

**EXPERIMENT-3**

**AIM:** To perform string operations:

* Merge 2 strings
* Reverse a string
* Find and replace substring

**CODE:**

#include <stdio.h>

#include <string.h>

//arjun yadav //

void merge\_strings(char string1[], char string2[], int a ,int b) {

    char merge[100];

    for(int i=0;i<a;i++){

        merge[i]=string1[i];

    }

    for(int j=0;j<b;j++){

        merge[j+a]=string2[j];

    }

    printf("Merged string is %s",merge);

}

void substring(char string1[],char string2[],char string3[]){

    {

    char ans[1000] = { 0 };

    int ans\_idx = 0;

    for (int i = 0; i < strlen(string1); i++) {

        int k = 0;

        if (string1[i] == string2[k] && i + strlen(string2) <= strlen(string1)) {

            int j;

            for (j = i; j < i + strlen(string2); j++) {

                if (string1[j] != string2[k]) {

                    break;

                }

                else {

                    k = k + 1;

                }

            }

            if (j == i + strlen(string2)) {

                for (int l = 0; l < strlen(string3); l++) {

                    ans[ans\_idx++] = string2[l];

                }

                i = j - 1;

            }

            else {

                ans[ans\_idx++] = string3[i];

            }

        }

    }

}

void reverse\_string(char str[]){

    int length = strlen(str);

    for (int i = 0; i < length / 2; i++) {

        char temp = str[i];

        str[i] = str[length - i - 1];

        str[length - i - 1] = temp;

    }

}

int main() {

    int choice;

    char string1[100], string2[100];

    int a=strlen(string1);

    int b=strlen(string2);

    int flag=1;

    while(flag){

        printf("Choose an option:\n");

        printf("1. Merge two strings\n");

        printf("2. Reverse a string\n");

        printf("3.Substring ");

        scanf("%d", &choice);

        getchar();

        switch (choice) {

            case 1:

                printf("Enter the first string: ");

                gets(string1);

                printf("Enter the second string: ");

                gets(string2);

                merge\_strings(string1, string2,a,b);

                break;

            case 2:

                printf("Enter the string to reverse: ");

                gets(string1);

                reverse\_string(string1);

                printf("Reversed string: %s\n", string1);

                break;

            case 3:

                printf("Enter the first string: ");

                gets(string1);

                printf("Enter the you want to find string: ");

                gets(string2);

                printf("Enter the you want to replace with string: ");

                gets(string3);

                substring(string1,string2,string3);

                break;

            case 4:

                flag=0;

                break;

            default:

                printf("Invalid choice.\n");

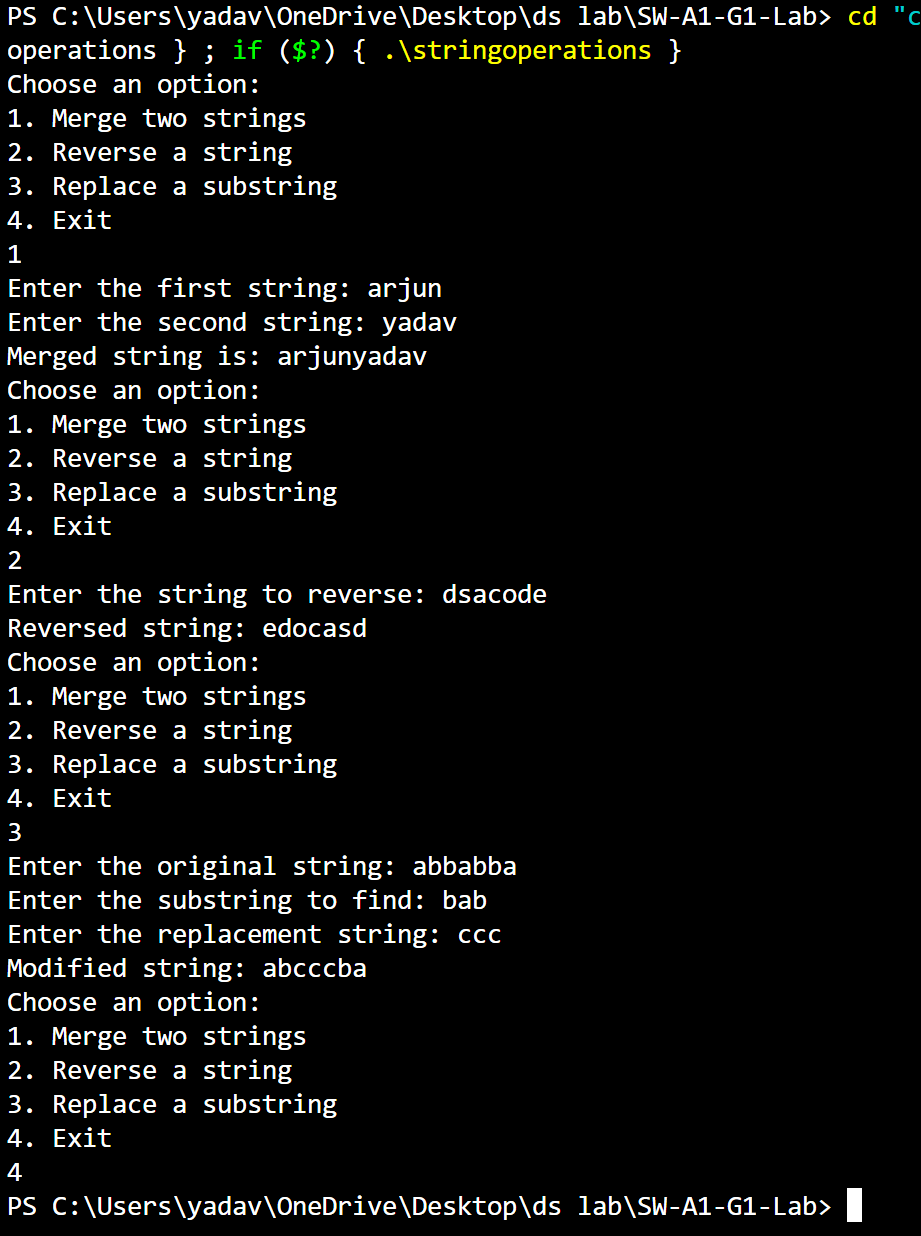
                break;

        }

    }

}

**OUTPUT:**

****