**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Shahabad Daulatpur, Bawana Road, Delhi-110042



**DEPARTMENT OF SOFTWARE ENGINEERING**

DATA STRUCTURES(SE-203)

Submitted To: Submitted By:

Mr. Ankur Narwal Anagh Bajpai

Department of Software Engineering 23/SE/022

| **S.No** | **Objective** | **Date** | **Sign** |
| --- | --- | --- | --- |
| 1. | Write a program in C to reverse an array | 21/08/24 |  |
| 2. | Write a Menu Driven C program to perform various operations on an array:   * Insert Element * Delete Element * Find Largest Element * Find Smallest Element | 04/09/24 |  |
| 3. | Write a Menu Driven C program to perform various operations on string:   * Merge 2 Strings * Reverse a String * Find and replace Substring | 09/10/24 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**EXPERIMENT No. 1**

**AIM:** Write a C Program to reverse an array of numbers.

**CODE:**

#include <stdio.h>

void ReverseArray(int arr[], int n) {

int start = 0;

int end = n - 1;

while (start < end) {

int temp = arr[start];

arr[start] = arr[end];

arr[end] = temp;

start++;

end--;

}

}

int main() {

int arr[] = {1, 2, 3, 4, 5};

int n = sizeof(arr) / sizeof(arr[0]);

printf("Original array: ");

for (int i = 0; i < n; i++) {

printf("%d ", arr[i]);

}

ReverseArray(arr, n);

printf("\nReversed array: ");

for (int i = 0; i < n; i++) {

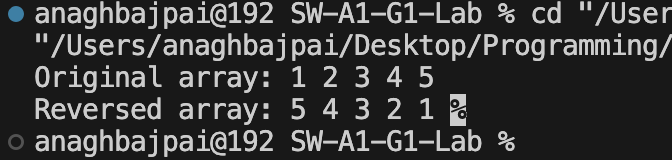
printf("%d ", arr[i]);

}

return 0;

}

**OUTPUT:**



**EXPERIMENT No. 2**

**AIM:** Write a Menu Driven C Program to perform various operations on array:

• Insert element

• Delete element(beginning, last, middle index)

• Find largest element

• Find smallest element

**CODE:**

#include <stdio.h>

#define MAX 100

void insert(int arr[], int \*top, int num){

if(\*top==MAX){

printf("Array Full\n");

}

else{

arr[\*top] = num;

\*top += 1;

}

}

void DeleteAtStart(int arr[], int \*top){

if(\*top==0){

printf("Array Empty\n");

}

else{

for(int i = 0; i<\*top;i++){

arr[i] = arr[i+1];

}

\*top -=1;

}

}

void DeleteAtEnd(int arr[], int \*top){

if(\*top==0){

printf("Array Empty\n");

}

else{

\*top -=1;

}

}

void DeleteAtMiddle(int arr[], int \*top){

int pos;

if(\*top==0){

printf("Array Empty\n");

}

else{

printf("Enter the position at which you want to delete: ");

scanf("%d", &pos);

if(pos<0||pos>\*top-1){

printf("Invalid Position\n");

}

else{

for(int i = pos; i<\*top;i++){

arr[i]=arr[i+1];

}

\*top -=1;

}

}

}

void findMax(int arr[],int top){

if(top==0){

printf("Array Empty\n");

}

else{

int max = arr[0];

for (int i = 1; i < top; i++) {

if (arr[i] > max) {

max = arr[i];

}

}

printf("Maximum value: %d\n", max);

}

}

void findMin(int arr[],int top){

if(top==0){

printf("Array Empty\n");

}

else{

int min = arr[0];

for (int i = 1; i < top; i++) {

if (arr[i] < min) {

min = arr[i];

}

}

printf("Maximum value: %d\n", min);

}

}

void display(int arr[], int n) {

if (n > 0) {

printf("Array elements: ");

for (int i = 0; i < n; i++) {

printf("%d ", arr[i]);

}

printf("\n");

} else {

printf("Array is empty\n");

}

}

int main(){

int arr[MAX];

int top = 0;

int choice, value;

do {

printf("\nMenu:\n");

printf("1. Insert\n");

printf("2. Delete at Start\n");

printf("3. Delete at Middle\n");

printf("4. Delete at End\n");

printf("5. Find Minimum\n");

printf("6. Find Maximum\n");

printf("7. Display Array\n");

printf("8. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter value to insert: ");

scanf("%d", &value);

insert(arr, &top, value);

break;

case 2:

DeleteAtStart(arr, &top);

break;

case 3:

DeleteAtMiddle(arr, &top);

break;

case 4:

DeleteAtEnd(arr, &top);

break;

case 5:

findMin(arr, top);

break;

case 6:

findMax(arr, top);

break;

case 7:

display(arr, top);

break;

case 8:

printf("Exiting...\n");

break;

default:

printf("Invalid choice\n");

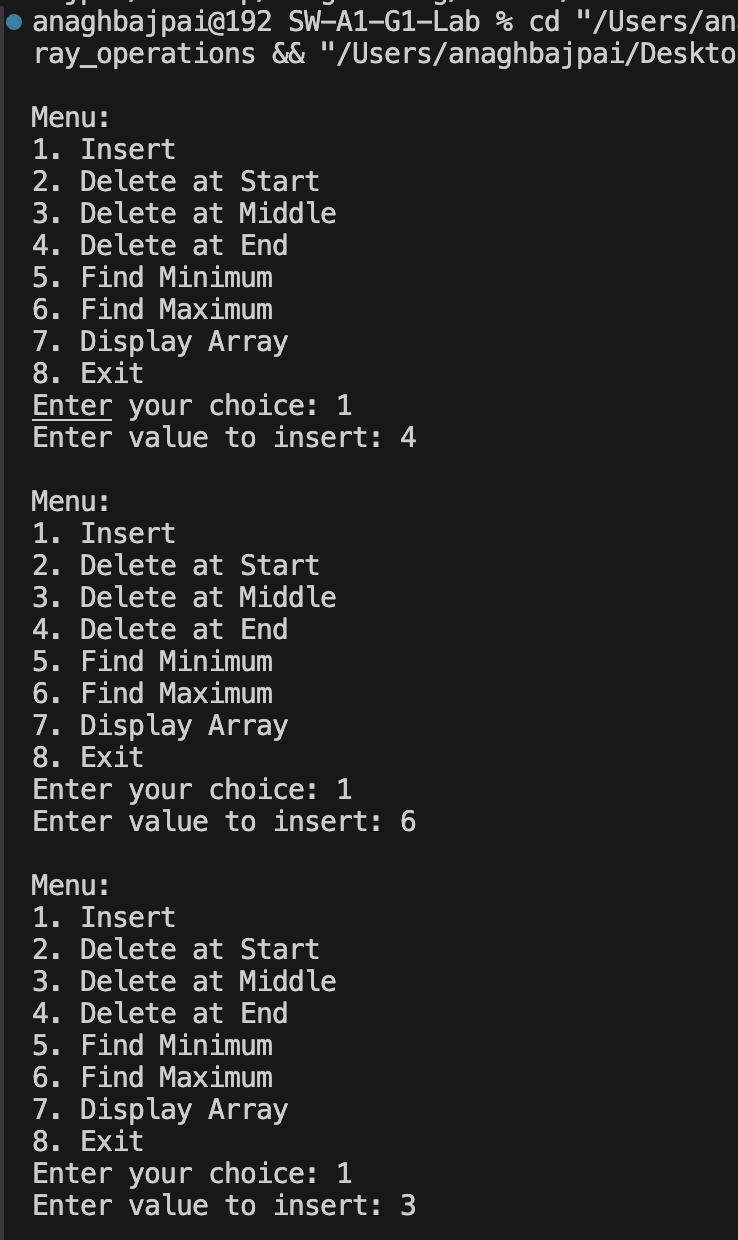
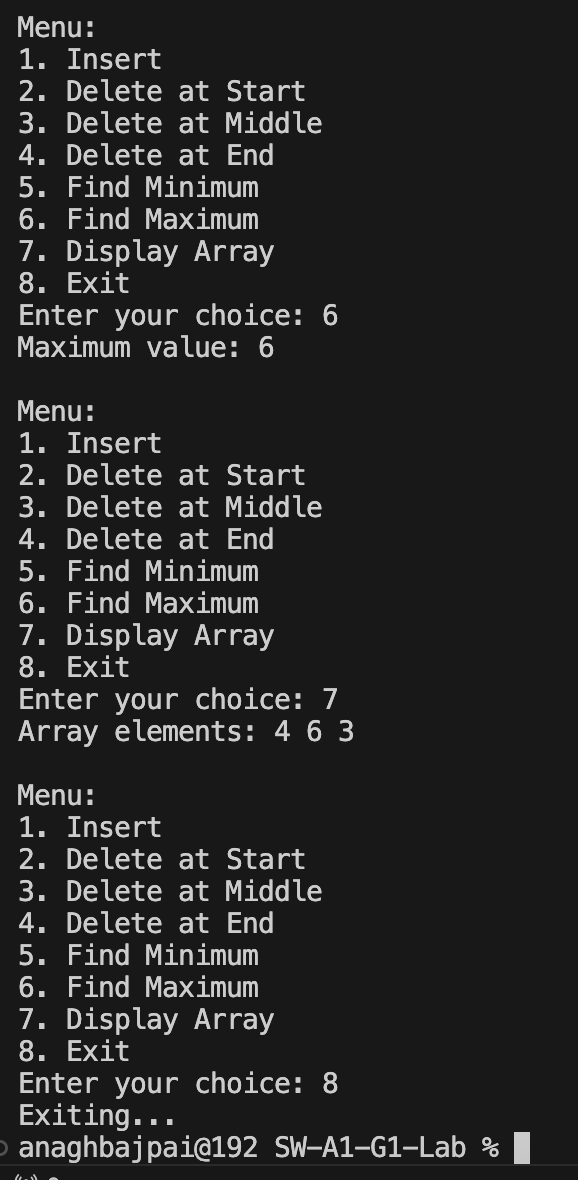
}

} while (choice != 8);

return 0;

}

**OUTPUT:**

** **

**EXPERIMENT No. 3**

**AIM:** Write a Menu Driven C Program to perform string operations:

• Merge 2 strings

• Reverse a string

• Find and replace substring

**CODE:**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

void mergeStrings(char str1[], char str2[]) {

char result[strlen(str1)+strlen(str2)+1];

int k=0;

for(int i=0;i<strlen(str1);i++){

result[k]=str1[i];

k++;

}

for(int i=0;i<strlen(str2);i++){

result[k]=str2[i];

k++;

}

result[k]='\0';

printf("Merged String: %s\n", result);

}

void reverseString(char str[]) {

int len = strlen(str);

char temp;

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

temp = str[i];

str[i] = str[len - i - 1];

str[len - i - 1] = temp;

}

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

}

void subString(char str[], char substr[], char newsubstr[]) {

char buffer[100];

char \*pos;

int index = 0;

int strLen = strlen(substr);

while ((pos = strstr(str, substr)) != NULL) {

int len = pos - str;

strncpy(buffer + index, str, len);

index += len;

strcpy(buffer + index, newsubstr);

index += strlen(newsubstr);

str += len + strLen;

}

strcpy(buffer + index, str);

strcpy(str, buffer);

printf("Updated string: %s\n", str);

}

int main(){

char str1[100], str2[100], str3[100];

int choice;

do {

printf("Menu:\n");

printf("1. Merge Two Strings\n");

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

printf("3. Find and Replace Substring\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

getchar();

switch (choice) {

case 1:

printf("Enter the first string: ");

fgets(str1, sizeof(str1), stdin);

str1[strcspn(str1, "\n")] = '\0';

printf("Enter the second string: ");

fgets(str2, sizeof(str2), stdin);

str2[strcspn(str2, "\n")] = '\0';

mergeStrings(str1, str2);

break;

case 2:

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

fgets(str1, sizeof(str1), stdin);

str1[strcspn(str1, "\n")] = '\0';

reverseString(str1);

break;

case 3:

printf("Enter the main string: ");

fgets(str1, 100, stdin);

str1[strcspn(str1, "\n")] = '\0';

printf("Enter the substring to find: ");

fgets(str2, 100, stdin);

str2[strcspn(str2, "\n")] = '\0';

printf("Enter the replacement substring: ");

fgets(str3, 100, stdin);

str3[strcspn(str3, "\n")] = '\0';

subString(str1, str2, str3);

break;

case 4:

printf("Exiting...\n");

break;

default:

printf("Invalid choice, please try again.\n");

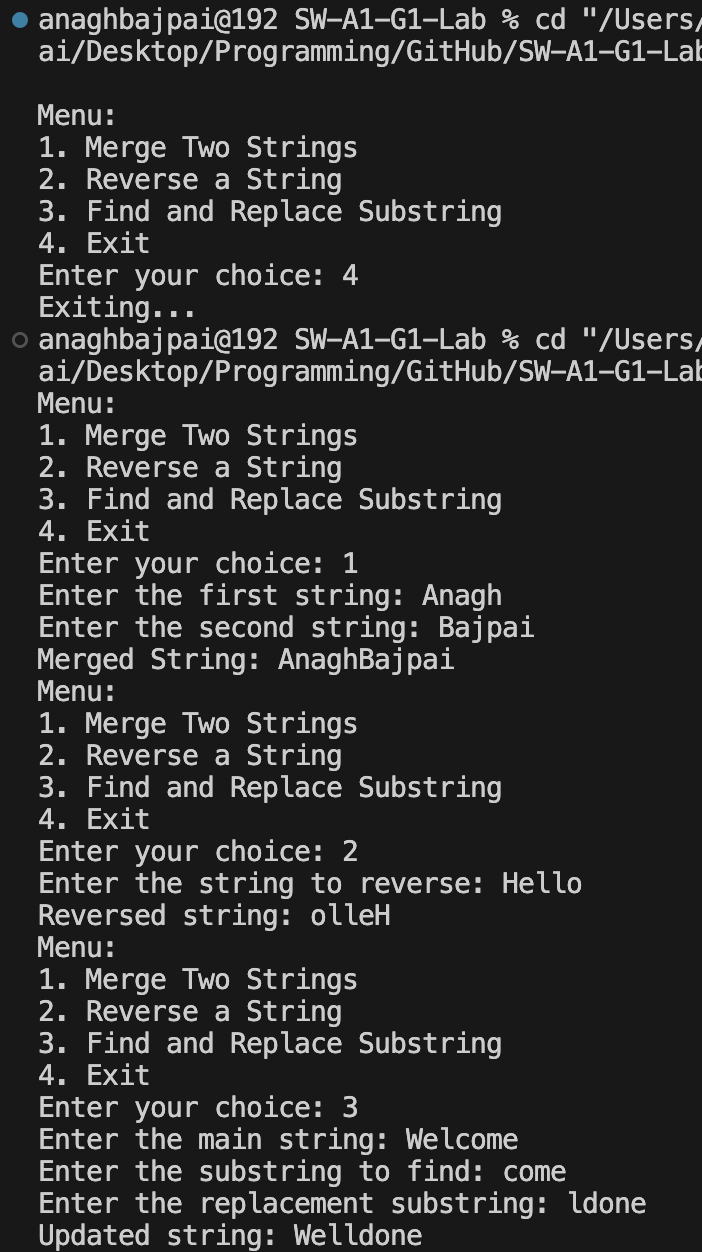
}

} while (choice != 4);

return 0;

}

**OUTPUT:**

****