**C PROGRAMMING ASSIGNMENT**

**GEU**

****

**BATCH**

**2023-2026**

**BCA(AI&DS)**

**SUBMITTED BY - AADARSH CHAUDHARY**

**SUBMITTED TO - MR. RISHI KUMAR**

**STUDENT ID - 23151260**

**ROLL NO. - 1**

**SECTION - I**

**ASSISTANT PROFESSOR - MS. POOJA CHAHAR**

1.Write a C program to exchange the values of two integer variables.

#include<stdio.h>

**int** main(**void**){

**int** A,B,temp;

printf("AADARSH CHAUDHARY\n");

printf("ENETR THE VALUE OF 'A' ");

scanf("%d",&A);

printf("ENTER THE VALUE OF 'B' ");

scanf("%d",&B);

//Swappin the values

temp = A;

A = B;

B = temp;

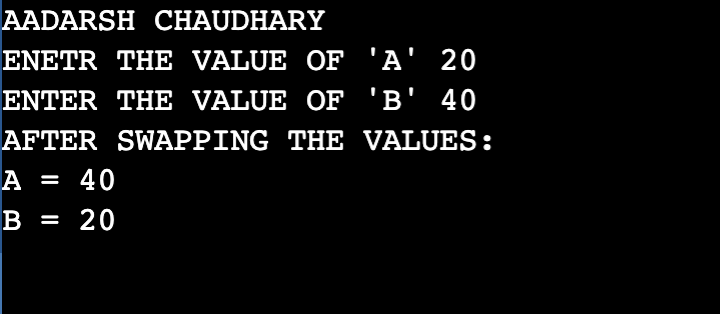
printf("AFTER SWAPPING THE VALUES:\n");

printf("A = %d\n",A);

printf("B = %d\n",B);

**return** 0;

}



2.Write a C program to find the roots of quadratic equations.

#include<stdio.h>

#include<math.h>

**int** main(**void**){

**int** a,b,c,D,root1,root2;

printf("AADARSH CHAUDHARY");

printf("\nEnter the coefficients:");

scanf("\n%d%d%d",&a,&b,&c);

D=b\*b-4\*a\*c;

**if**(D>0){

root1=(-b+sqrt(D))/(2\*a);

root2=(-b-sqrt(D))/(2\*a);

printf("\nRoots are real and different.");

printf("\nroot1=%d",root1);

printf("\nroot2=%d",root2);

}

**else** **if**(D==0){

root1=root2=(-b)/(2\*a);

printf("Roots are real and same");

printf("\nroot1=%d",root1);

printf("\nroot2=%d",root2);

}

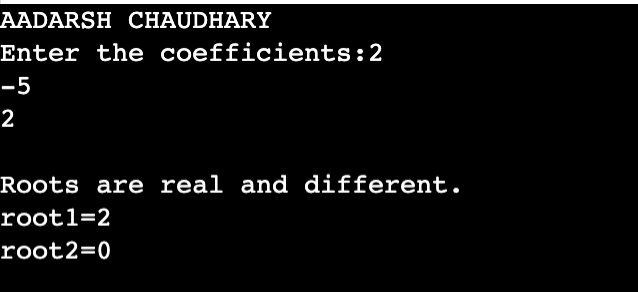
**else**{

printf("No real roots exist");

}

**return** 0;

}



3.Write a C program to check whether the entered integer is a palindrome.

#include <stdio.h>

int main() {

int num, reversedNum = 0, remainder, originalNum;

printf("Enter an integer: ");

scanf("%d", &num);

originalNum = num;

// Reversing the number

while (num != 0) {

remainder = num % 10;

reversedNum = reversedNum \* 10 + remainder;

num /= 10;

}

// Checking if the reversed number is equal to the original number

if (originalNum == reversedNum) {

printf("%d is a palindrome.", originalNum);

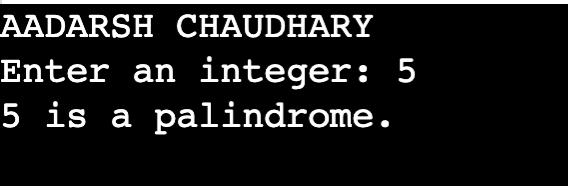
} else {

printf("%d is not a palindrome.", originalNum);

}

return 0;

}



4.Write a C program to sort the elements of an array in ascending order.

#include <stdio.h>

void bubbleSort(int arr[], int size) {

int i, j, temp;

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

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

if (arr[j] > arr[j+1]) {

// Swapping the elements

temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

}

int main() {

int arr[] = {5, 2, 8, 1, 3};

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

int i;

printf("AADARSH CHAUDHARY\n");

printf("Original array: ");

for (i = 0; i < size; i++) {

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

}

bubbleSort(arr, size);

printf("\nSorted array in ascending order: ");

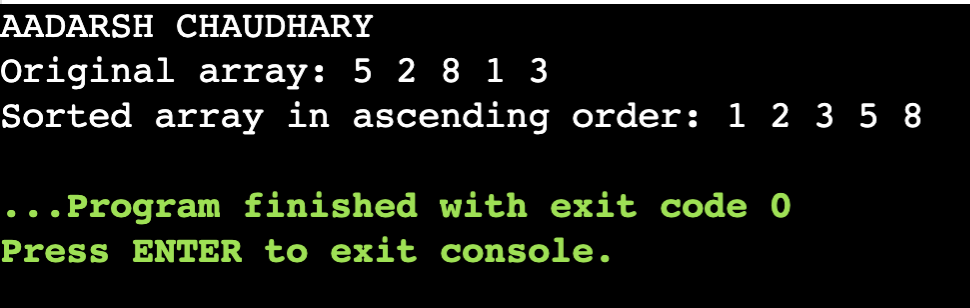
for (i = 0; i < size; i++) {

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

}

return 0;

}



5.Write a C program to search for an element in an array. Display the position of the element.

#include <stdio.h>

int searchElement(int arr[], int size, int element) {

int i;

for (i = 0; i < size; i++) {

if (arr[i] == element) {

return i; // Found the element, returning its position

}

}

return -1; // Element not found

}

int main() {

int arr[] = {5, 2, 8, 1, 3};

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

int element, position;

printf("AADARSH CHAUDHARY\n");

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

scanf("%d", &element);

position = searchElement(arr, size, element);

if (position != -1) {

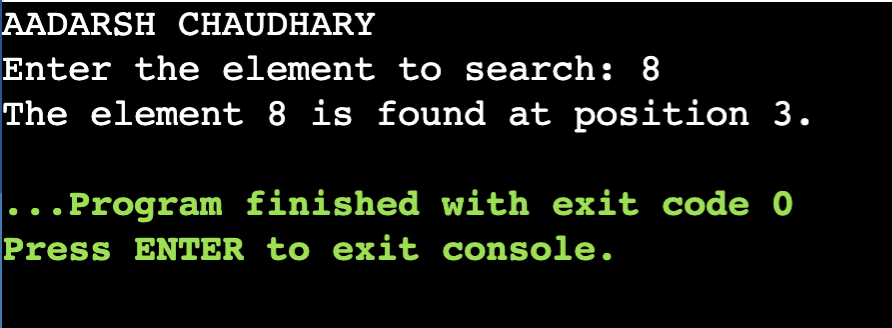
printf("The element %d is found at position %d.", element, position + 1);

} else {

printf("The element %d is not found in the array.", element);

}

return 0;

}

6.Consider two matrices of the size m and n. Implement matrix multiplication operation and display results using functions. Write three functions 1) Read matrix elements 2) Matrix Multiplication 3) Print matrix elements

#include <stdio.h>

void readMatrix(int matrix[][100], int m, int n) {

printf("AADARSH CHAUDHARY\n");

printf("Enter matrix elements:\n");

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

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

printf("Enter element at position [%d][%d]: ", i + 1, j + 1);

scanf("%d", &matrix[i][j]);

}

}

}

void multiplyMatrices(int mat1[][100], int mat2[][100],

int result[][100], int m, int n, int p) {

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

for (int j = 0; j < p; j++) {

result[i][j] = 0;

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

result[i][j] += mat1[i][k] \* mat2[k][j];

}

}

}

}

void printMatrix(int matrix[][100], int m, int n) {

printf("Matrix:\n");

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

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

printf("%d\t", matrix[i][j]);

}

printf("\n");

}

}

int main() {

int m, n, p;

printf("Enter dimensions of the first matrix (m x n): ");

scanf("%d %d", &m, &n);

printf("Enter dimensions of the second matrix (n x p): ");

scanf("%d %d", &n, &p);

int mat1[100][100], mat2[100][100], result[100][100];

readMatrix(mat1, m, n);

readMatrix(mat2, n, p);

multiplyMatrices(mat1, mat2, result, m, n, p);

printf("\nFirst ");

printMatrix(mat1, m, n);

printf("\nSecond ");

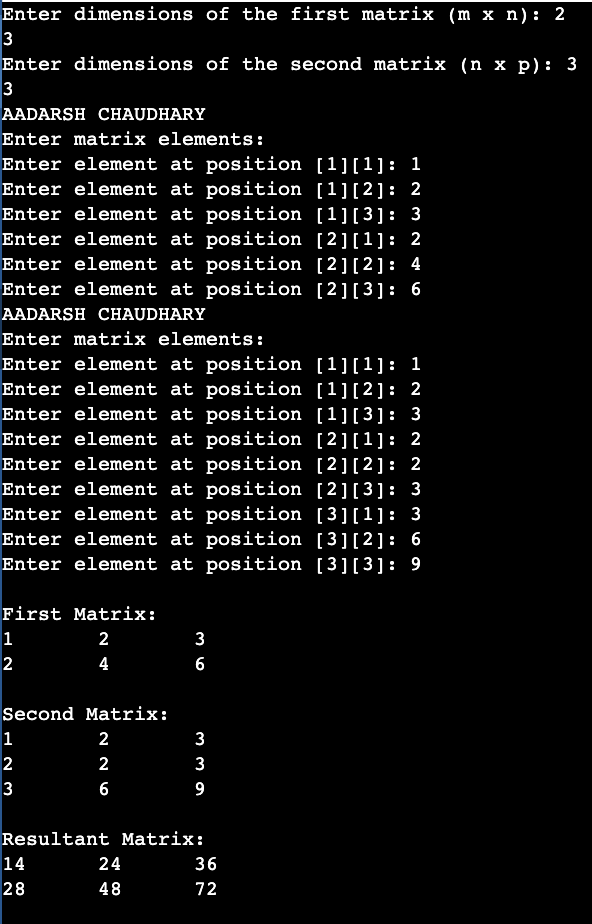
printMatrix(mat2, n, p);

printf("\nResultant ");

printMatrix(result, m, p);

return 0;

}



­­7.Consider two strings S1 and S2. Develop a C Program for the following operations.a) Display a concatenated output of S1 and S2b) Count the number of characters and empty spaces in S1and S2.

#include <stdio.h>

#include <string.h>

int main() {

char s1[100], s2[100], concatenated[200];

int countCharacters, countSpaces;

printf("AADARSH CHAUDHARY\n");

printf("Enter string S1: ");

gets(s1);

printf("Enter string S2: ");

gets(s2);

// Concatenate S1 and S2

strcpy(concatenated, s1);

strcat(concatenated, s2);

printf("Concatenated output of S1 and S2: %s\n", concatenated);

// Count characters and empty spaces in S1 and S2

countCharacters = strlen(s1) + strlen(s2);

countSpaces = 0;

for (int i = 0; i < strlen(s1); i++) {

if (s1[i] == ' ') {

countSpaces++;

}

}

for (int i = 0; i < strlen(s2); i++) {

if (s2[i] == ' ') {

countSpaces++;

}

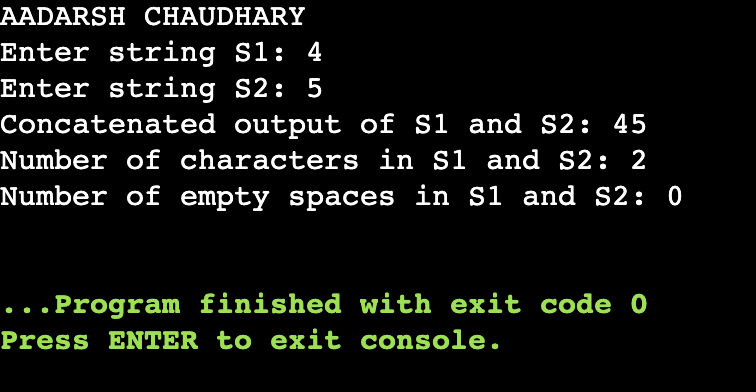
}

printf("Number of characters in S1 and S2: %d\n", countCharacters);

printf("Number of empty spaces in S1 and S2: %d\n", countSpaces);

return 0;

}



8.Consider details of a bank account with the fields account number, account holder’s name, and balance. Write a program to read 10 people’s details and display the record with the highest bank balance.

#include <stdio.h>

#include <string.h>

struct BankAccount {

int accountNumber;

char accountHolderName[50];

double balance;

};

int main() {

struct BankAccount accounts[10];

double maxBalance = 0;

int maxBalanceIndex = 0;

printf("AADARSH CHAUDHARY\n");

// Read details of 10 people's bank accounts

for (int i = 0; i < 10; i++) {

printf("Enter details for person %d:\n", i + 1);

printf("Account number: ");

scanf("%d", &accounts[i].accountNumber);

printf("Account holder's name: ");

scanf(" %[^\n]s", accounts[i].accountHolderName);

printf("Balance: ");

scanf("%lf", &accounts[i].balance);

printf("\n");

}

// Find the record with the highest bank balance

for (int i = 0; i < 10; i++) {

if (accounts[i].balance > maxBalance) {

maxBalance = accounts[i].balance;

maxBalanceIndex = i;

}

}

// Display the record with the highest bank balance

printf("Record with the highest bank balance:\n");

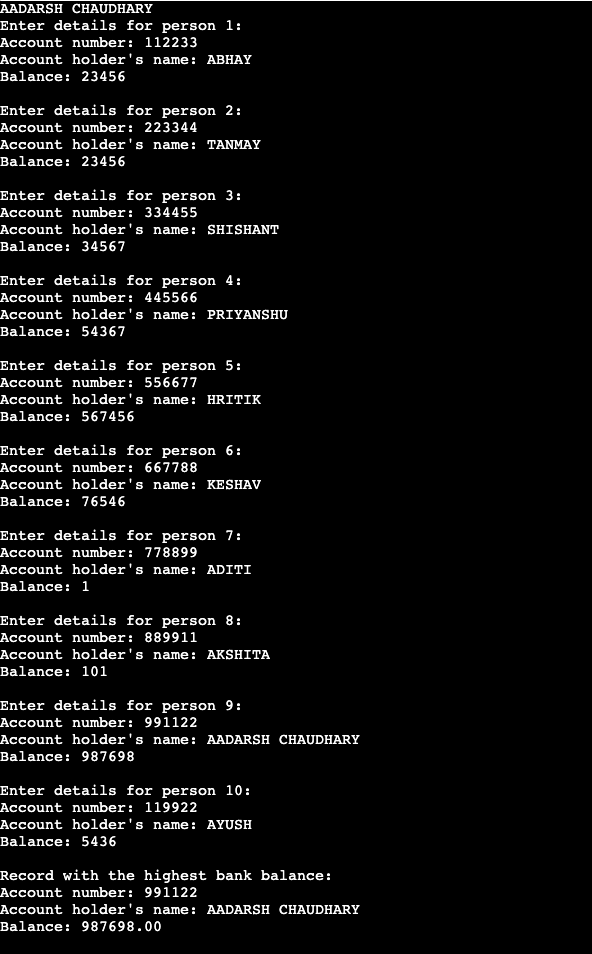
printf("Account number: %d\n", accounts[maxBalanceIndex].accountNumber);

printf("Account holder's name: %s\n", accounts[maxBalanceIndex].accountHolderName);

printf("Balance: %.2lf\n", accounts[maxBalanceIndex].balance);

return 0;

}



9.Write a C program to demonstrate the use of & and \* operators using pointers. Create and free a memory location for an integer. Display the address and data stored at the location.

#include <stdio.h>

#include <stdlib.h>

int main() {

printf("AADARSH CHAUDHARY\n");

int\* ptr = (int\*)malloc(sizeof(int));

if (ptr == NULL) {

printf("Memory allocation failed. Exiting...\n");

return 1;

}

printf("Enter an integer: ");

scanf("%d", ptr);

printf("\nMemory Location Information:\n");

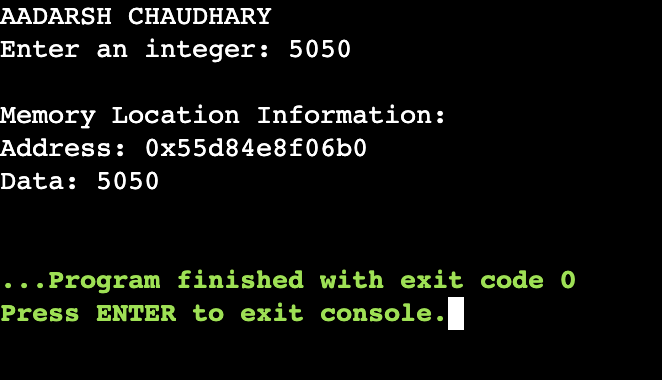
printf("Address: %p\n", (void\*)ptr);

printf("Data: %d\n", \*ptr);

free(ptr);

return 0;

}



10.Write a program to create a file called student.txt and store information about a student in terms of roll no, age, and marks.

#include <stdio.h>

int main() {

// Open the file in write mode

FILE \*file = fopen("student.txt", "w");

if (file == NULL) {

printf("Unable to create the file.");

return 1;

}

// Get student information from the user

int rollNo, age;

float marks;

printf("AADARSH CHAUDHARY\n");

printf("Enter the roll number: ");

scanf("%d", &rollNo);

fprintf(file, "Roll Number: %d\n", rollNo);

printf("Enter the age: ");

scanf("%d", &age);

fprintf(file, "Age: %d\n", age);

printf("Enter the marks: ");

scanf("%f", &marks);

fprintf(file, "Marks: %.2f\n", marks);

// Close the file

fclose(file);

printf("Student information has been stored in 'student.txt'.\n");

return 0;

}