**ASSIGNMENT**

**OF C PROGRAMMING**



**SUBMITTED BY: SUBMITTED TO:**

**ASHUTOSH THAPA MR. Rishi Kumar**

**ROLL NO.:**

**47**

**COURSE:**

**BCA(Hons) AI&DS**

**ID:**

**23151179**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **LIST OF PROGRAMS** | **Faculty Sign** | **Remarks** |
| **1** | **Write a C program to exchange the values of two integer variables.** |  |  |
| **2** | **Write a C program to find the roots of quadratic equations.** |  |  |
| **3** | **Write a C program to check whether the entered integer is a palindrome.** |  |  |
| **4** | **Write a C program to sort the elements of an array in ascending order.** |  |  |
| **5** | **Write a C program to search for an element in an array. Display the position of the element.** |  |  |
| **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** |  |  |
| **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.** |  |  |
| **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.** |  |  |
| **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.** |  |  |
| **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.** |  |  |

**C PROGRAM**

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

**#include <stdio.h>**

**void exchangeValues(int \*a, int \*b) {**

**int temp = \*a;**

**\*a = \*b;**

**\*b = temp;**

**}**

**int main() {**

**int num1, num2;**

**printf("Enter the first integer: ");**

**scanf("%d", &num1);**

**printf("Enter the second integer: ");**

**scanf("%d", &num2);**

**printf("\nBefore exchange:\n");**

**printf("First integer: %d\n", num1);**

**printf("Second integer: %d\n", num2);**

**exchangeValues(&num1, &num2);**

**printf("\nAfter exchange:\n");**

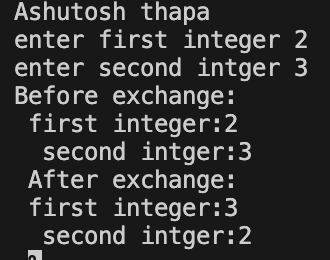
**printf("First integer: %d\n", num1);**

**printf("Second integer: %d\n", num2);**

**return 0;**

**}**

**Out put:**

****

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

**#include <stdio.h>**

**#include <math.h>**

**void findRoots(float a, float b, float c) {**

**float discriminant = b \* b - 4 \* a \* c;**

**if (discriminant > 0) {**

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

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

**printf("Root 1: %.2f\n", root1);**

**printf("Root 2: %.2f\n", root2);**

**} else if (discriminant == 0) {**

**// Real and equal roots**

**float root = -b / (2 \* a);**

**printf("Root 1 = Root 2: %.2f\n", root);**

**} else {**

**// Complex roots**

**float realPart = -b / (2 \* a);**

**float imaginaryPart = sqrt(-discriminant) / (2 \* a);**

**printf("Root 1: %.2f + %.2fi\n", realPart, imaginaryPart);**

**printf("Root 2: %.2f - %.2fi\n", realPart, imaginaryPart);**

**}**

**}**

**int main() {**

**float a, b, c;**

**printf("Enter coefficient a: ");**

**scanf("%f", &a);**

**printf("Enter coefficient b: ");**

**scanf("%f", &b);**

**printf("Enter coefficient c: ");**

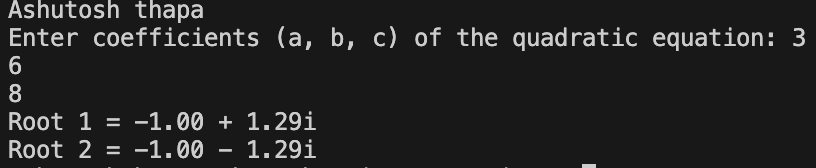
**scanf("%f", &c);**

**findRoots(a, b, c);**

**return 0;**

**}**

**Out put:**

****

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

**#include <stdio.h>**

**int isPalindrome(int num) {**

**int originalNum = num;**

**int reversedNum = 0, remainder;**

**while (num > 0) {**

**remainder = num % 10;**

**reversedNum = reversedNum \* 10 + remainder;**

**num /= 10;**

**}**

**return originalNum == reversedNum;**

**}**

**int main() {**

**int number;**

**printf("Enter an integer: ");**

**scanf("%d", &number);**

**if (isPalindrome(number)) {**

**printf("%d is a palindrome.\n", number);**

**} else {**

**printf("%d is not a palindrome.\n", number);**

**}**

**return 0;**

**}**

**Out put:**

**A black background with white text

Description automatically generated**

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

**#include <stdio.h>**

**void bubbleSort(int arr[], int n) {**

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

**for (int j = 0; j < n - i - 1; j++) {**

**// Swap if the element found is greater**

**// than the next element**

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

**int temp = arr[j];**

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

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

**}**

**}**

**}**

**}**

**void printArray(int arr[], int size) {**

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

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

**}**

**printf("\n");**

**}**

**int main() {**

**int arr[] = {64, 34, 25, 12, 22, 11, 90};**

**int n = sizeof(arr)]);**

**printf("Original array: ");**

**printArray(arr, n);**

**bubbleSort(arr, n);**

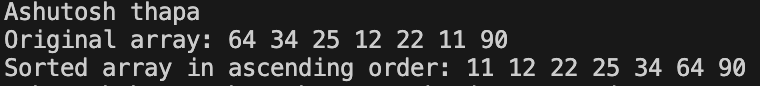
**printf("Sorted array in ascending order: ");**

**printArray(arr, n);**

**return 0;**

**}**

**Out put:**

****

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

**#includestdio.h>  
int main(){**

**int a[100], size, i , item, found;**

**printf(“Ashutosh Thapa\n”);**

**printf(“enter the size of array”);**

**scanf(“%d”,&size);**

**printf(“enter the elements”);**

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

**scanf(“%d”,&a[i]);**

**}**

**printf(“enter the element which you want to search”);**

**scanf(“%d”,&item);**

**found=0;**

**for(i=0;i<size;i++){  
if(a[i]==item){**

**found=1;**

**break;**

**}**

**}**

**if(found==1){**

**printf(“position is %d”,i+1);**

**}**

**else{**

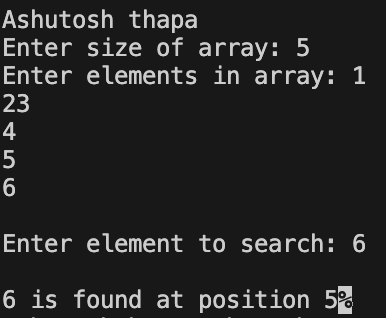
**printf(“not found”);**

**}**

**return 0;**

**}**

**out put:**

****

**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 getMatrixElements(int matrix[][10], int row, int column) {**

**printf("\nEnter elements: \n");**

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

**for (int j = 0; j < column; ++j) {**

**printf("Enter a%d%d: ", i + 1, j + 1);**

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

**}**

**}**

**}**

**void multiplyMatrices(int first[][10],**

**int second[][10],**

**int result[][10],**

**int r1, int c1, int r2, int c2) {**

**for (int i = 0; i < r1; ++i) {**

**for (int j = 0; j < c2; ++j) {**

**result[i][j] = 0;**

**}**

**}**

**for (int i = 0; i < r1; ++i) {**

**for (int j = 0; j < c2; ++j) {**

**for (int k = 0; k < c1; ++k) {**

**result[i][j] += first[i][k] \* second[k][j];**

**}**

**}**

**}**

**}**

**void display(int result[][10], int row, int column) {**

**printf("\nOutput Matrix:\n");**

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

**for (int j = 0; j < column; ++j) {**

**printf("%d ", result[i][j]);**

**if (j == column - 1)**

**printf("\n");**

**}**

**}**

**}**

**int main() {**

**int first[10][10], second[10][10], result[10][10], r1, c1, r2, c2;**

**printf(“Ashutosh thapa\n”):**

**printf("Enter rows and column for the first matrix: ");**

**scanf("%d %d", &r1, &c1);**

**printf("Enter rows and column for the second matrix: ");**

**scanf("%d %d", &r2, &c2);**

**while (c1 != r2) {**

**printf("Error! Enter rows and columns again.\n");**

**printf("Enter rows and columns for the first matrix: ");**

**scanf("%d%d", &r1, &c1);**

**printf("Enter rows and columns for the second matrix: ");**

**scanf("%d%d", &r2, &c2);**

**}**

**getMatrixElements(first, r1, c1);**

**getMatrixElements(second, r2, c2);**

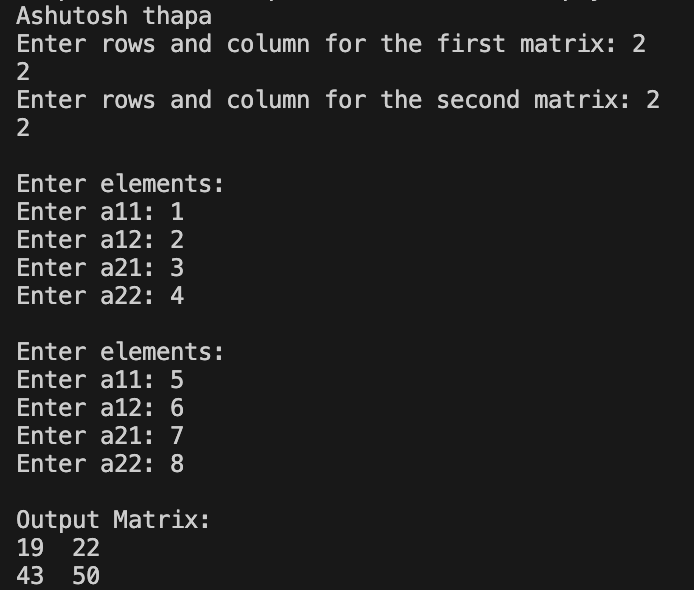
**multiplyMatrices(first, second, result, r1, c1, r2, c2);**

**display(result, r1, c2);**

**return 0;**

**}**

**Out put:**

****

**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];**

**printf(“Ashutosh Thapa\n”);**

**printf("Enter string S1: ");**

**gets(S1);**

**printf("Enter string S2: ");**

**gets(S2);**

**printf("\nConcatenated strings: %s%s\n", S1, S2);**

**int lenS1 = strlen(S1);**

**int spacesS1 = 0;**

**for (int i = 0; i < lenS1; i++) {**

**if (S1[i] == ' ') {**

**spacesS1++;**

**}**

**}**

**S2);**

**int spacesS2 = 0;**

**for (int i = 0; i < lenS2; i++) {**

**if (S2[i] == ' ') {**

**spacesS2++;**

**}**

**}**

**printf("\nNumber of characters in S1: %d\n", lenS1);**

**printf("Number of spaces in S1: %d\n", spacesS1);**

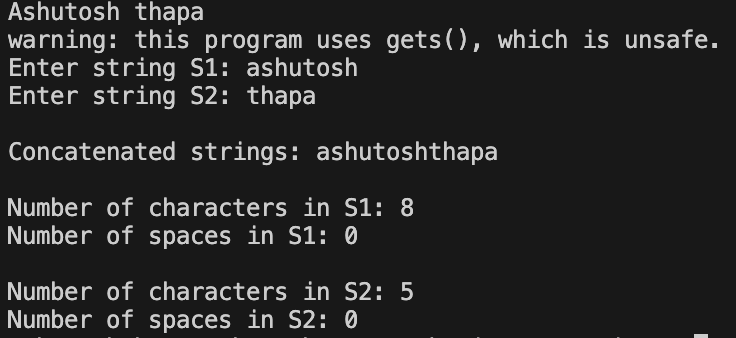
**printf("\nNumber of characters in S2: %d\n", lenS2);**

**printf("Number of spaces in S2: %d\n", spacesS2);**

**return 0;**

**}**

**Out put:**

****

**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>**

**// Structure to represent a bank account**

**struct BankAccount {**

**int accountNumber;**

**char accountHolderName[100];**

**float balance;**

**};**

**int main() {**

**struct BankAccount accounts[10];**

**printf(“Ashutosh thapa\n”);**

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

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

**printf("Account Number: ");**

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

**printf("Account Holder's Name: ");**

**scanf("%s", accounts[i].accountHolderName); printf("Balance: ");**

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

**printf("\n");**

**}**

**int maxIndex = 0;**

**for (int i = 1; i < 10; i++) {**

**if (accounts[i].balance > accounts[maxIndex].balance) {**

**maxIndex = i;**

**}**

**}**

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

**printf("Account Number: %d\n", accounts[maxIndex].accountNumber);**

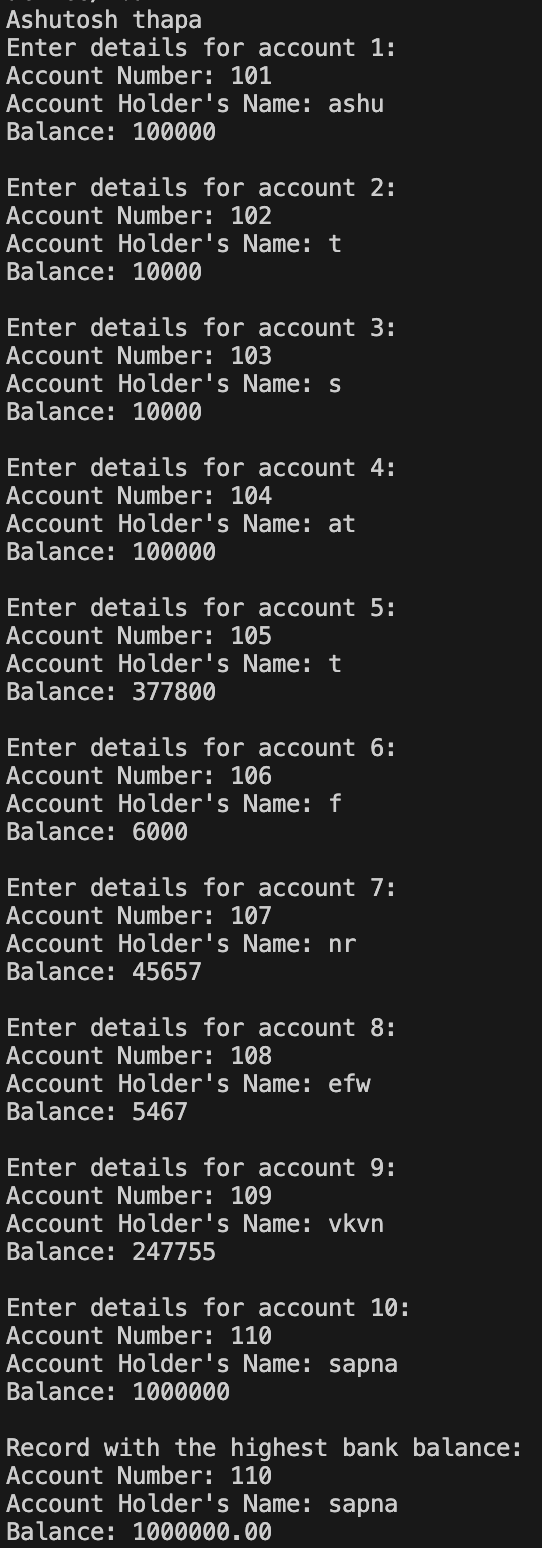
**printf("Account Holder's Name: %s\n", accounts[maxIndex].accountHolderName);**

**printf("Balance: %.2f\n", accounts[maxIndex].balance);**

**return 0;**

**}**

**Out put:**

****

**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() {**

**int \*ptr;**

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

**printf(“Ashutosh Thapa\n”);**

**if (ptr == NULL) {**

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

**return 1;**

**}**

**printf("Enter an integer: ");**

**scanf("%d", ptr);**

**printf("\nAddress of the allocated memory: %p\n", (void \*)ptr);**

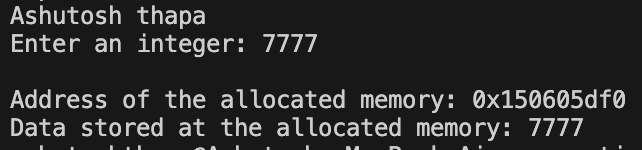
**printf("Data stored at the allocated memory: %d\n", \*ptr);**

**free(ptr);**

**return 0;**

**}**

**Out put:**

****

**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>**

**struct Student {**

**int rollNumber;**

**int age;**

**float marks;**

**};**

**int main() {**

**FILE \*filePointer;**

**struct Student student;**

**filePointer = fopen("student.txt", "w");**

**if (filePointer == NULL) {**

**printf("Error opening the file. Exiting...\n");**

**return 1;**

**}**

**printf(“Ashutosh Thapa\n”);**

**printf("Enter student information:\n");**

**printf("Roll Number: ");**

**scanf("%d", &student.rollNumber);**

**printf("Age: ");**

**scanf("%d", &student.age);**

**printf("Marks: ");**

**scanf("%f", &student.marks);**

**fprintf(filePointer, "Roll Number: %d\n", student.rollNumber);**

**fprintf(filePointer, "Age: %d\n", student.age);**

**fprintf(filePointer, "Marks: %.2f\n", student.marks);**

**fclose(filePointer);**

**printf("Student information has been written to student.txt.\n");**

**return 0;**

**}**

**Out put:**

**A black background with white text

Description automatically generated**