**DAY-1 LAB PROGRAMS**

**QUE-1**

#include <stdio.h>

int fibonacci(int n) {

if (n <= 1) {

return n;

}

return fibonacci(n - 1) + fibonacci(n - 2);

}

int main() {

int n, i;

printf("Enter the number of terms: ");

scanf("%d", &n);

printf("Fibonacci series: ");

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

printf("%d ", fibonacci(i));

}

printf("\n");

return 0;

}

**QUE-2**

**#include <stdio.h>**

**#include <math.h>**

**int main() {**

**int num, originalNum, remainder, n = 0;**

**float result = 0.0;**

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

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

**originalNum = num;**

**while (originalNum != 0) {**

**originalNum /= 10;**

**++n;**

**}**

**originalNum = num;**

**while (originalNum != 0) {**

**remainder = originalNum % 10;**

**result += pow(remainder, n);**

**originalNum /= 10;**

**}**

**if ((int)result == num) {**

**printf("%d is an Armstrong number.\n", num);**

**} else {**

**printf("%d is not an Armstrong number.\n", num);**

**}**

**return 0;**

**}**

**QUE-3**

**#include <stdio.h>**

**int gcd(int a, int b) {**

**if (b == 0) {**

**return a;**

**}**

**return gcd(b, a % b);**

**}**

**int main() {**

**int num1, num2;**

**printf("Enter two integers: ");**

**scanf("%d %d", &num1, &num2);**

**printf("The GCD of %d and %d is %d\n", num1, num2, gcd(num1, num2));**

**return 0;**

**}**

**QUE-4**

**#include <stdio.h>**

**int main() {**

**int n;**

**printf("Enter the number of elements in the array: ");**

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

**int arr[n];**

**printf("Enter the elements of the array:\n");**

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

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

**}**

**int largest = arr[0];**

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

**if (arr[i] > largest) {**

**largest = arr[i];**

**}**

**}**

**printf("The largest element in the array is %d\n", largest);**

**return 0;**

**}**

**QUE-5**

**#include <stdio.h>**

**int factorial(int n) {**

**if (n == 0 || n == 1) {**

**return 1;**

**}**

**return n \* factorial(n - 1);**

**}**

**int main() {**

**int num;**

**printf("Enter a number to find its factorial: ");**

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

**printf("The factorial of %d is %d\n", num, factorial(num));**

**return 0;**

**}**

**QUE-6**

**#include <stdio.h>**

**int isPrime(int n, int i) {**

**if (n < 2) {**

**return 0;**

**}**

**if (i == n) {**

**return 1;**

**}**

**if (n % i == 0) {**

**return 0;**

**}**

**return isPrime(n, i + 1);**

**}**

**int main() {**

**int num;**

**printf("Enter a number to check if it is prime: ");**

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

**if (isPrime(num, 2)) {**

**printf("%d is a prime number\n", num);**

**} else {**

**printf("%d is not a prime number\n", num);**

**}**

**return 0;**

**}**

**QUE-7**

**#include <stdio.h>**

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

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

**int minIndex = i;**

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

**if (arr[j] < arr[minIndex]) {**

**minIndex = j;**

**}**

**}**

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

**arr[minIndex] = arr[i];**

**arr[i] = temp;**

**}**

**}**

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

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

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

**}**

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

**}**

**int main() {**

**int n;**

**printf("Enter the number of elements in the array: ");**

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

**int arr[n];**

**printf("Enter the elements of the array:\n");**

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

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

**}**

**selectionSort(arr, n);**

**printf("Sorted array: \n");**

**printArray(arr, n);**

**return 0;**

**}**

**QUE-8**

**#include <stdio.h>**

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

**int i, j;**

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

**// Last i elements are already in place**

**for (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) {**

**int i;**

**for (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)/sizeof(arr[0]);**

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

**printArray(arr, n);**

**bubbleSort(arr, n);**

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

**printArray(arr, n);**

**return 0;**

**}**

**QUE-9**

**#include <stdio.h>**

**#include <time.h>**

**#define MAX\_SIZE 100**

**void multiplyMatrices(int A[MAX\_SIZE][MAX\_SIZE], int B[MAX\_SIZE][MAX\_SIZE], int C[MAX\_SIZE][MAX\_SIZE], int m, int n, int p) {**

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

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

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

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

**C[i][j] += A[i][k] \* B[k][j];**

**}**

**}**

**}**

**}**

**int main() {**

**int A[MAX\_SIZE][MAX\_SIZE], B[MAX\_SIZE][MAX\_SIZE], C[MAX\_SIZE][MAX\_SIZE];**

**int m, n, p;**

**clock\_t start, end;**

**double cpu\_time\_used;**

**printf("Enter dimensions of matrix A (m x n): ");**

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

**printf("Enter dimensions of matrix B (n x p): ");**

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

**printf("Enter elements of matrix A:\n");**

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

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

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

**}**

**}**

**printf("Enter elements of matrix B:\n");**

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

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

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

**}**

**}**

**start = clock();**

**multiplyMatrices(A, B, C, m, n, p);**

**end = clock();**

**cpu\_time\_used = ((double) (end - start)) / CLOCKS\_PER\_SEC;**

**printf("Resultant matrix C:\n");**

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

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

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

**}**

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

**}**

**printf("Time taken to multiply matrices: %f seconds\n", cpu\_time\_used);**

**printf("Time complexity estimation: O(%d \* %d \* %d) = O(%d)\n", m, n, p, m \* n \* p);**

**return 0;**

**}**

**QUE-10**

**#include <stdio.h>**

**#include <stdbool.h>**

**#include <string.h>**

**bool isPalindrome(char str[], int left, int right) {**

**if (left >= right) {**

**return true;**

**}**

**return (str[left] == str[right]) && isPalindrome(str, left + 1, right - 1);**

**}**

**int main() {**

**char str[100];**

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

**scanf("%s", str);**

**if (isPalindrome(str, 0, strlen(str) - 1)) {**

**printf("%s is a palindrome.\n", str);**

**} else {**

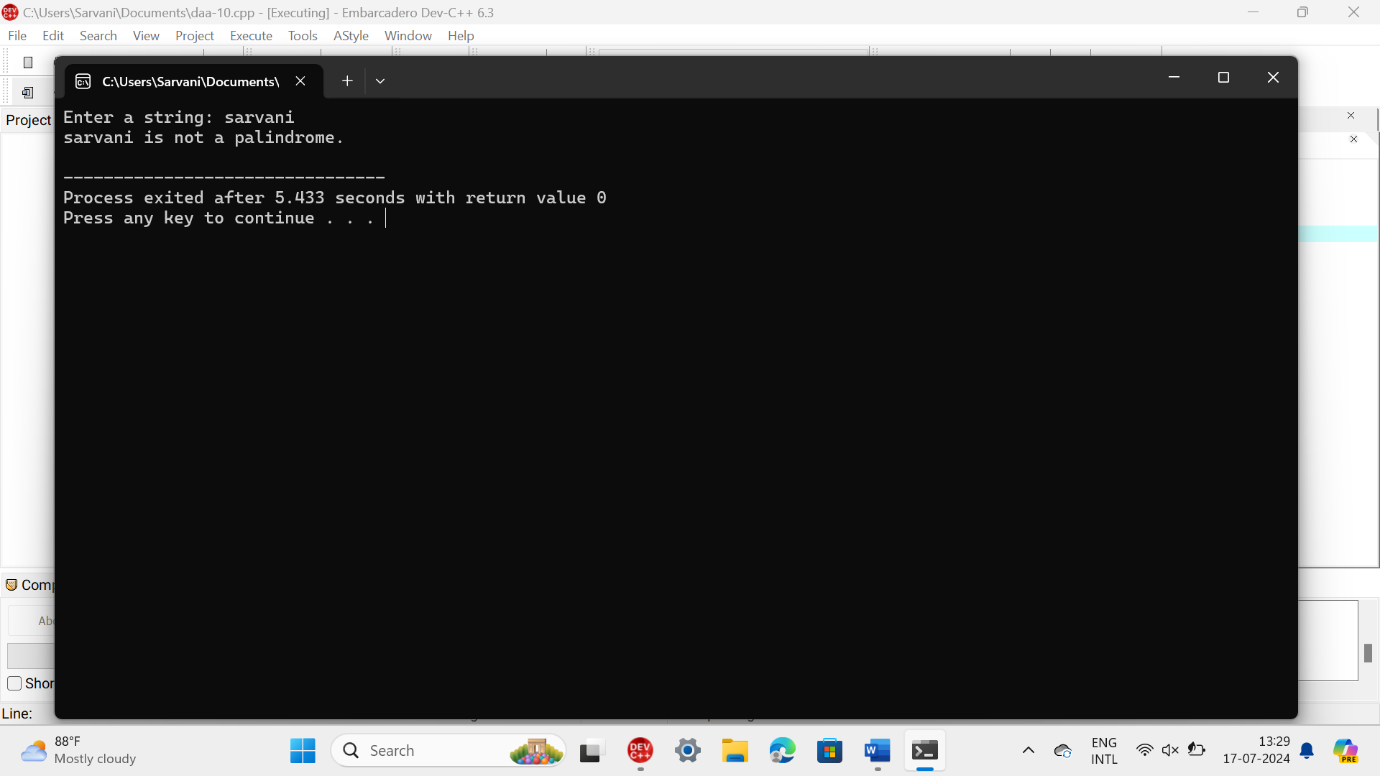
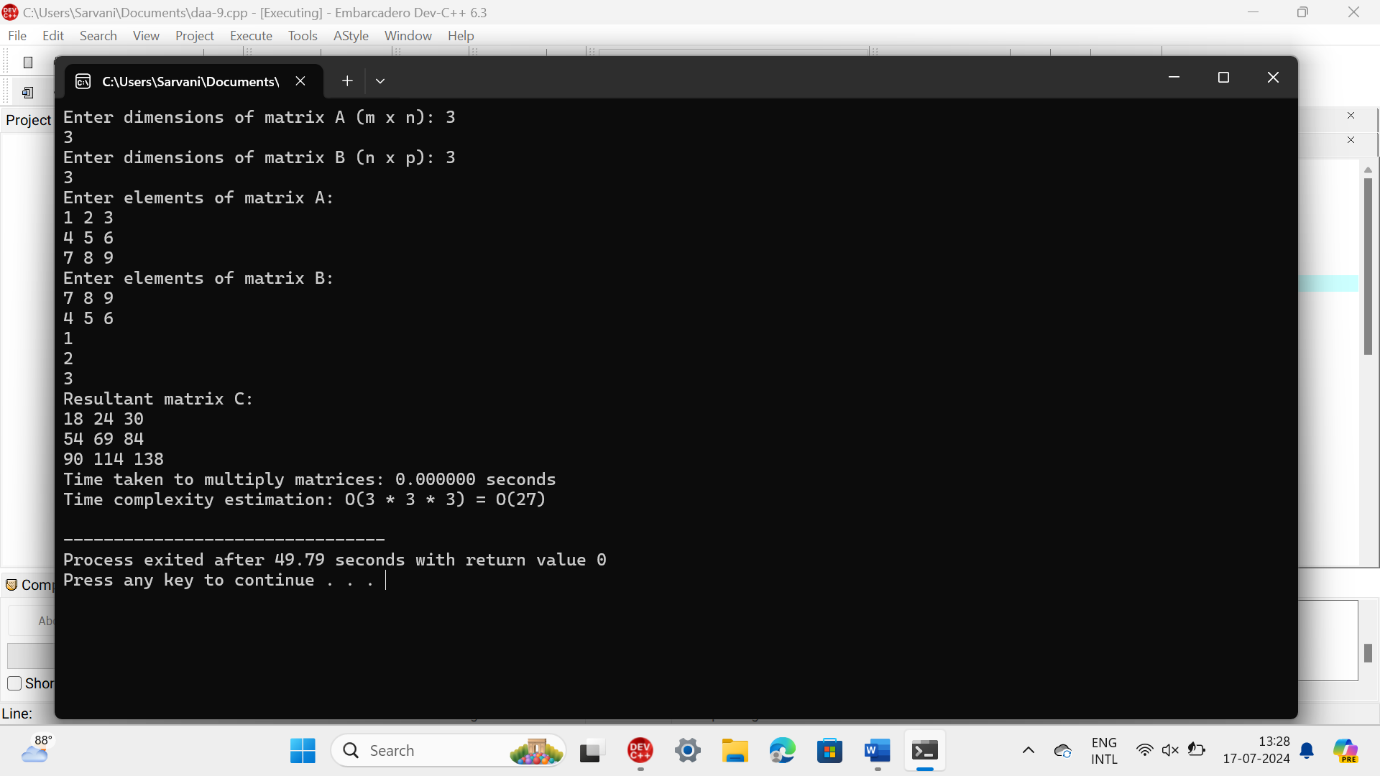
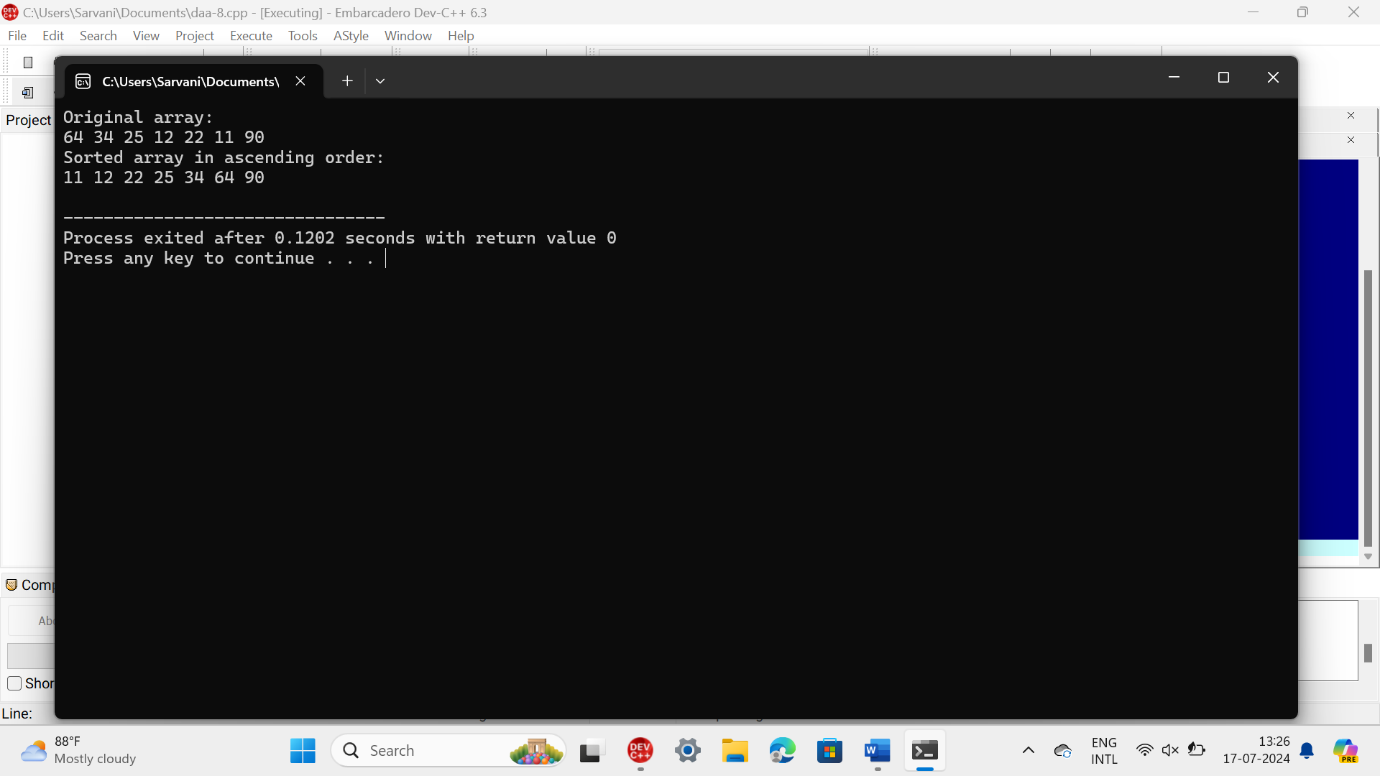
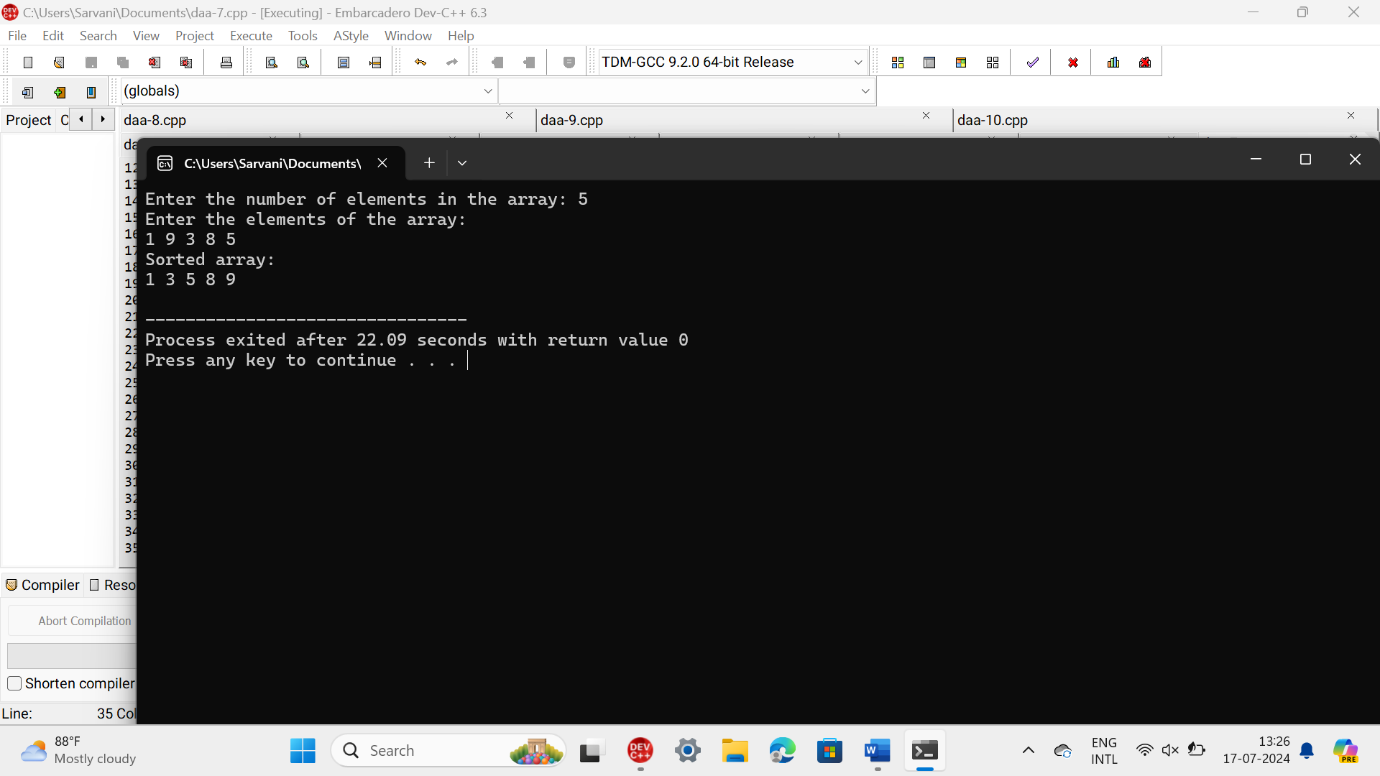
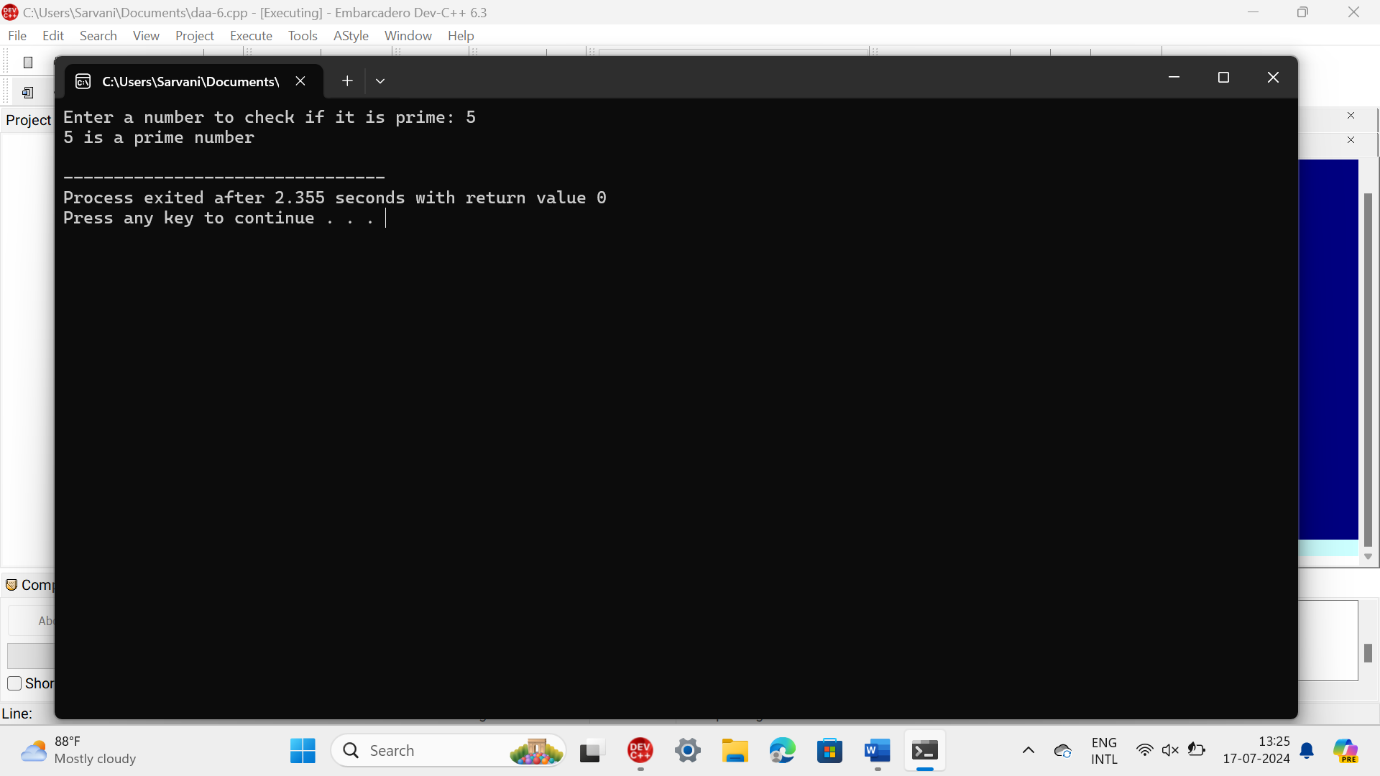
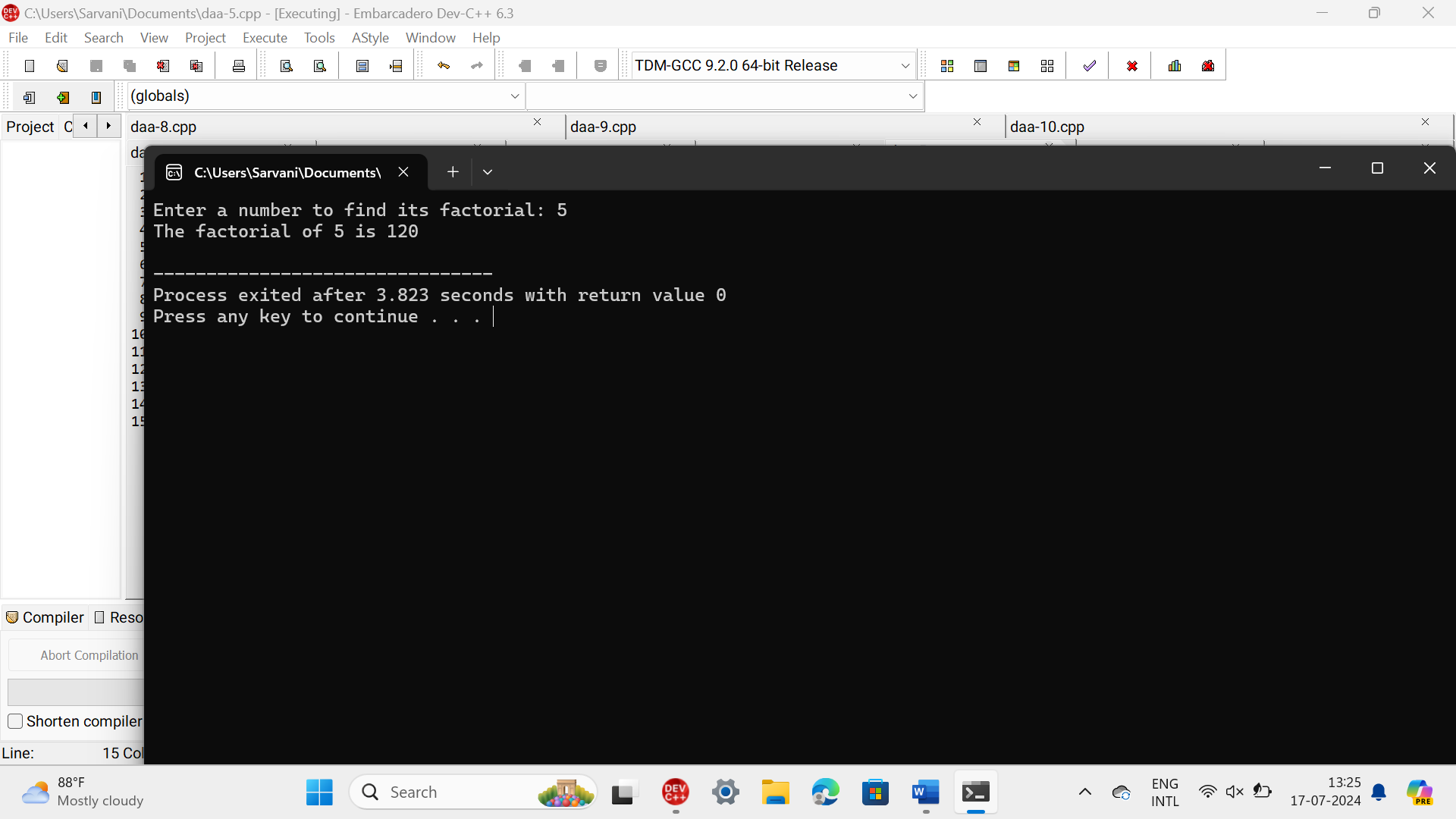
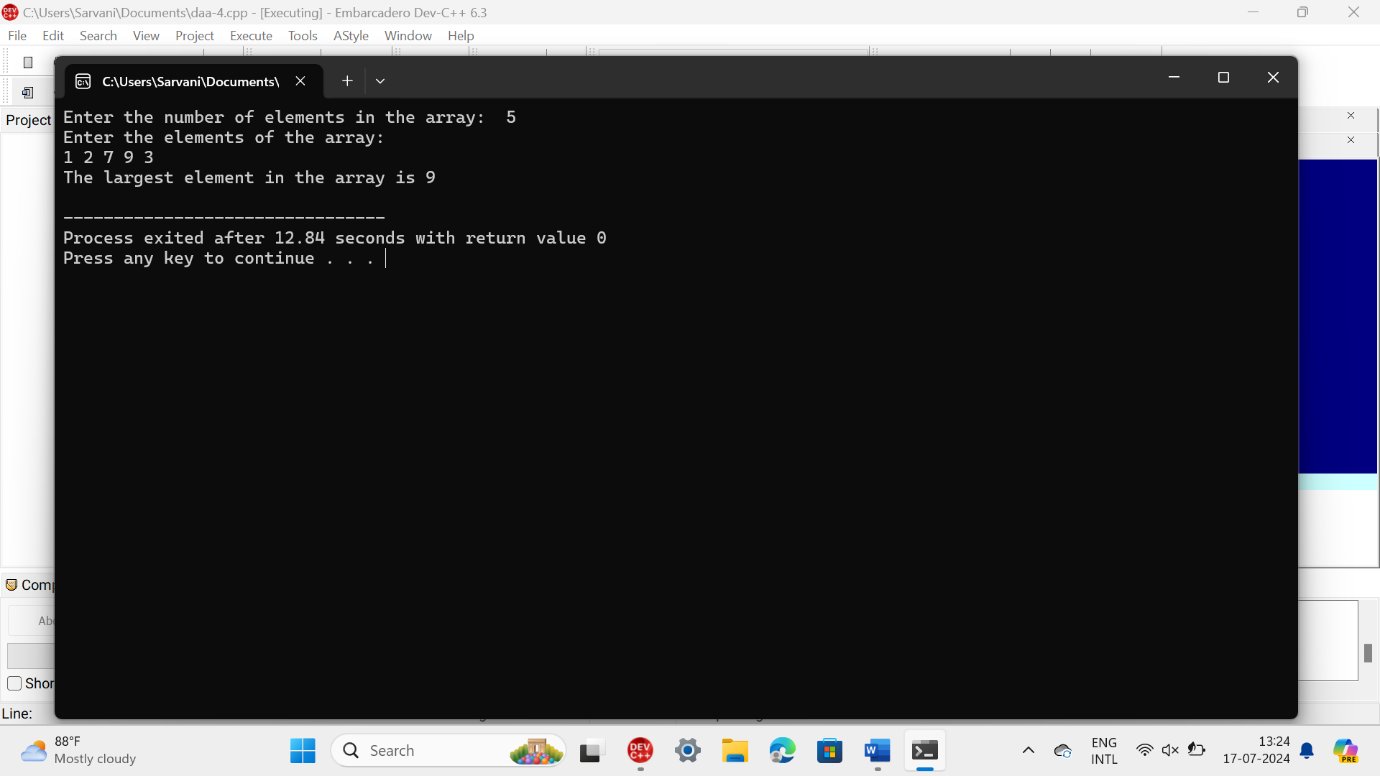
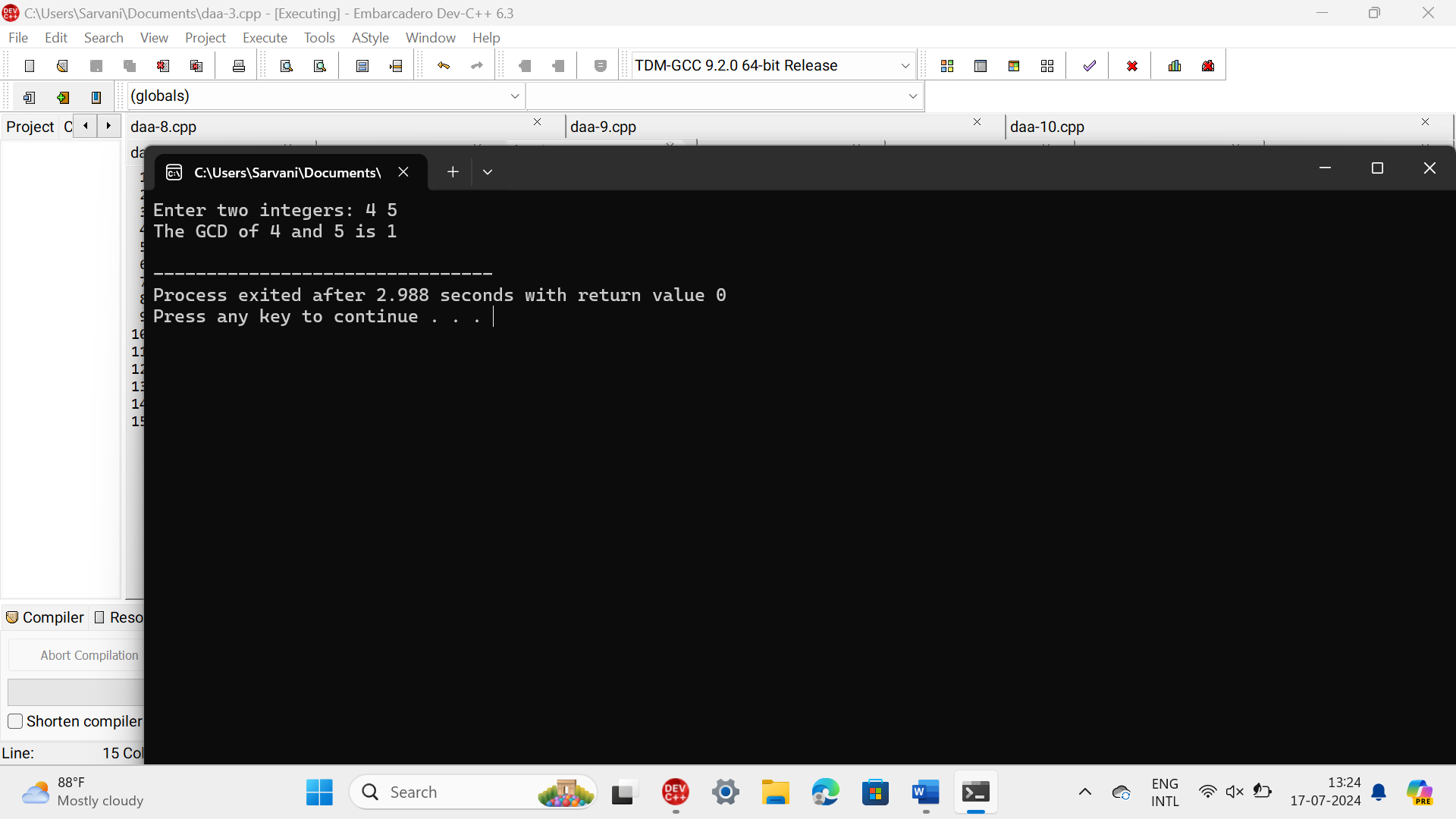
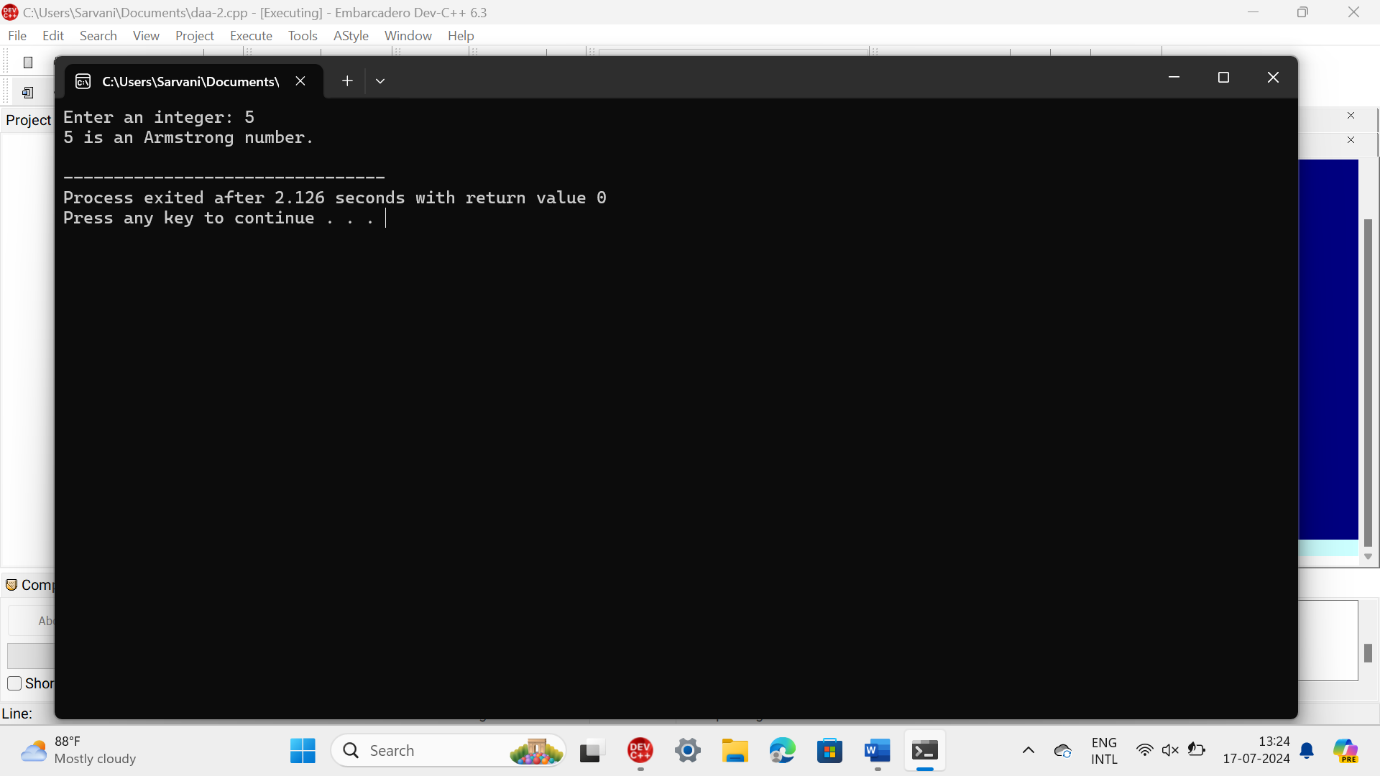
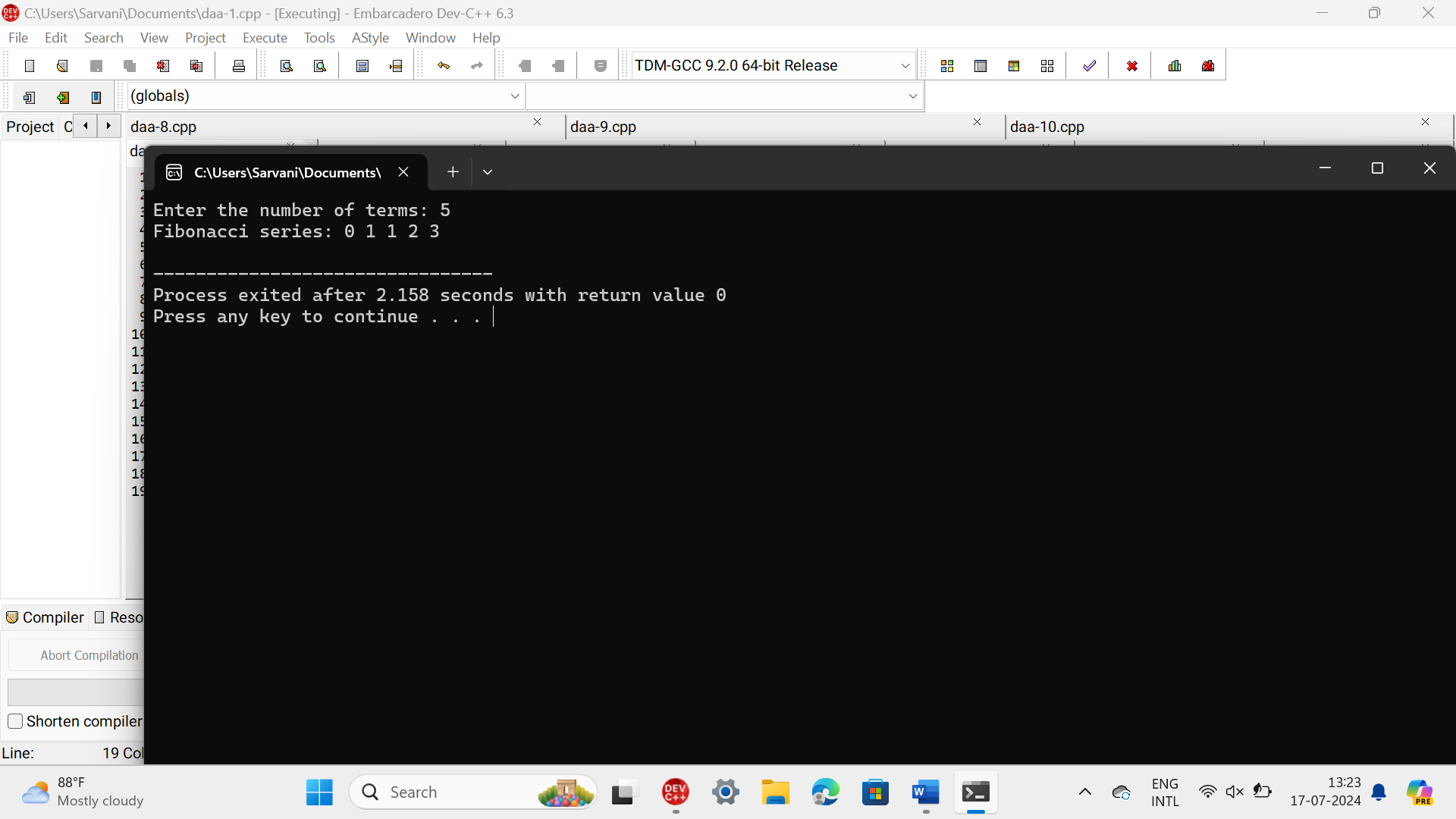
**printf("%s is not a palindrome.\n", str);**

**}**

**return 0;**

**}**

**OUTPUTS**

****