**Daily Assessment**

**Muhammad Ammar Chaudhry**

**SU92-BSCSM-F23-269**

**BSCS-1G**

**----------------------------------------------------------------------------------------------------**

**Q # 91:**

#include <iostream>

using namespace std;

int callCount = 0;

void countMe() {

callCount++;

cout << "Function has been called " << callCount << " time"<<((callCount==1)? ".":"s.") << endl;

}

int main() {

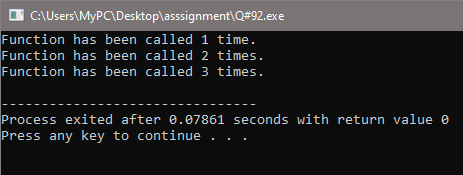
countMe();

countMe();

countMe();

return 0;

}



**Q # 92:**

#include <iostream>

using namespace std;

void countMeStatic() {

static int callCount = 0;

callCount++;

cout << "Function has been called " << callCount << " time"<<((callCount==1)? ".":"s.") << endl;

}

int main() {

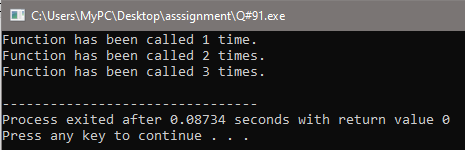
countMeStatic();

countMeStatic();

countMeStatic();

return 0;

}



**Q # 93:**

#include <iostream>

using namespace std;

void calcTotal( double arrayA[], int arrayB[], double arrayC[], int size) {

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

arrayC[i] = arrayA[i] - arrayB[i];

}

}

int main() {

const int maxSize = 100;

double arrayA[maxSize];

int arrayB[maxSize];

double arrayC[maxSize];

int size;

cout << "Enter the size of the arrays: ";

cin >> size;

cout << "Enter " << size << " double numbers for array A:\n";

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

cin >> arrayA[i];

}

cout << "Enter " << size << " integer numbers for array B:\n";

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

cin >> arrayB[i];

}

calcTotal(arrayA, arrayB, arrayC, size);

cout << "Result array C:\n";

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

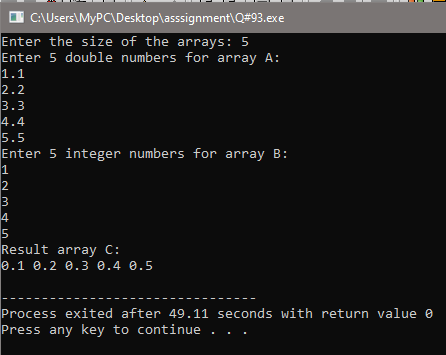
cout << arrayC[i] << " ";

}

cout << endl;

return 0;

}



**Q # 94:**

#include <iostream>

using namespace std;

int ARRAY\_SIZE = 15;

bool valueExists( int arr[], int size, int value) {

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

if (arr[i] == value) {

return true;

}

}

return false;

}

int main() {

int userValues[ARRAY\_SIZE];

int count = 0;

cout << "Enter 15 positive integer values:\n";

while (count < ARRAY\_SIZE) {

int newValue;

cout << "Enter value #" << (count + 1) << ": ";

cin >> newValue;

if (newValue > 0 && !valueExists(userValues, count, newValue)) {

userValues[count] = newValue;

count++;

} else if (newValue > 0) {

cout << "Sorry: Value Already Exist\n";

} else {

cout << "Please enter a positive integer.\n";

}

}

cout << "Values stored in the array:\n";

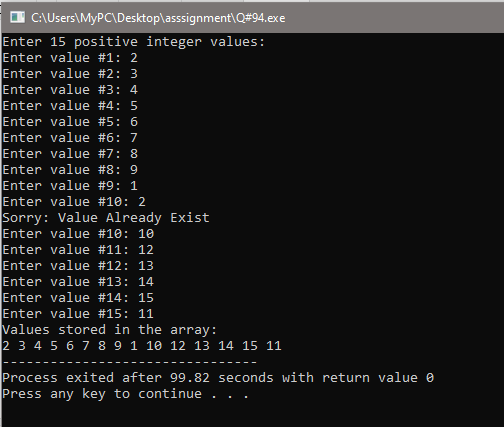
for (int i = 0; i < ARRAY\_SIZE; ++i) {

cout << userValues[i] << " ";

}

return 0;

}



**Q # 95:**

#include <iostream>

using namespace std;

int findMin( int arr[], int size) {

if (size == 1) {

return arr[0];

}

int minVal = arr[0];

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

if (arr[i] < minVal) {

minVal = arr[i];

}

}

return minVal;

}

int main() {

int size;

cout<<"Enter the size of the Array:";

cin>>size;

int userArray[size];

cout << "Enter " << size << " integers:\n";

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

cin >> userArray[i];

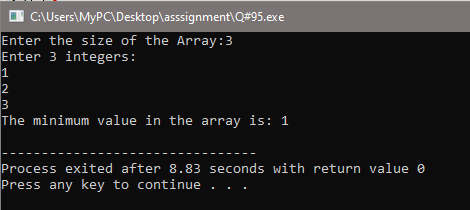
}

int minValue = findMin(userArray, size);

cout << "The minimum value in the array is: " << minValue << endl;

return 0;

}



**Q # 96:**

#include <iostream>

#include <cmath>

using namespace std;

bool isPrime(int number) {

if (number <= 1) {

return false;

}

for (int i = 2; i <= sqrt(number); ++i) {

if (number % i == 0) {

return false;

}

}

return true;

}

int main() {

const int maxPrimes = 10;

int primeNumbers[maxPrimes];

int count = 0;

cout << "Enter positive integers. The program will store prime numbers and stop after storing 10." << endl;

while (count < maxPrimes) {

int userInput;

cout << "Enter a number: ";

cin >> userInput;

if (isPrime(userInput)) {

primeNumbers[count] = userInput;

++count;

}

}

cout << "\nPrime numbers stored in the array:" << endl;

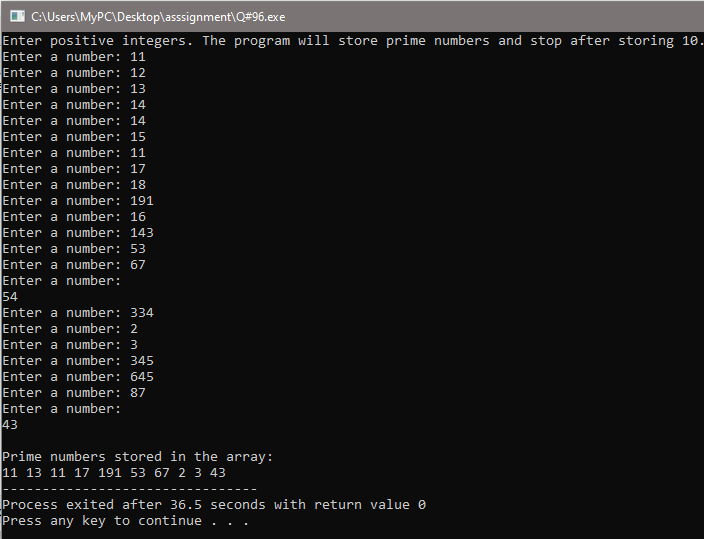
for (int i = 0; i < maxPrimes; ++i) {

cout << primeNumbers[i] << " ";

}

return 0;

}



**Q # 97:**

#include <iostream>

using namespace std;

int lastIndexOccurrence( int arr[], int size, int x) {

int lastIndex = -1;

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

if (arr[i] == x) {

lastIndex = i;

}

}

return lastIndex;

}

int main() {

int maxSize = 100;

int A[maxSize];

int size, x;

cout << "Enter the size of the array: ";

cin >> size;

cout << "Enter " << size << " integers for the array:\n";

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

cin >> A[i];

}

cout << "Enter the integer 'x' to search for: ";

cin >> x;

int lastIndex = lastIndexOccurrence(A, size, x);

if (lastIndex != -1) {

cout << "The last occurrence of '" << x << "' is at index: " << lastIndex << endl;

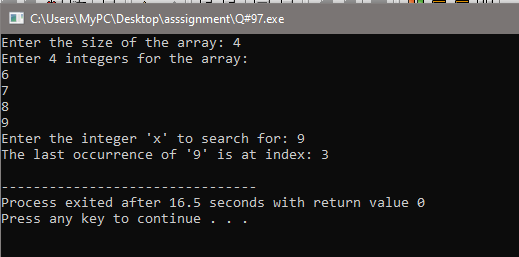
} else {

cout << "'" << x << "' not found in the array." << endl;

}

return 0;

}



**Q # 98:**

#include <iostream>

using namespace std;

int main() {

const int rows = 3;

const int columns = 5;

int matrix[rows][columns];

cout << "Enter elements for a 3x5 matrix:\n";

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

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

cout << "Enter element at position (" << i + 1 << ", " << j + 1 << "): ";

cin >> matrix[i][j];

}

}

cout << "\nMatrix form:\n";

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

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

cout << matrix[i][j] << " ";

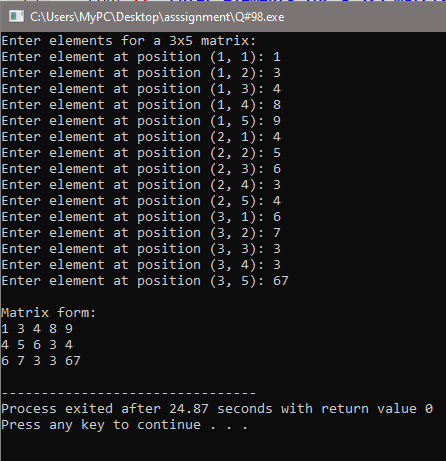
}

cout << endl;

}

return 0;

}



**Q # 99:**

#include <iostream>

using namespace std;

int main() {

const int maxSize = 100;

char inputString[maxSize];

cout << "Enter a string: ";

cin.getline(inputString, maxSize);

int length = 0;

while (inputString[length] != '\0') {

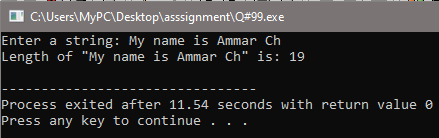
length++;

}

cout << "Length of \"" << inputString << "\" is: " << length << endl;

return 0;

}



**Q # 100:**

#include <iostream>

#include <cstring>

using namespace std;

int main() {

const int maxSize = 100;

char userName[maxSize];

cout << "Enter your name: ";

cin.getline(userName, maxSize);

int length = strlen(userName);

cout << "Reverse of \"" << userName << "\" is: ";

for (int i = length - 1; i >= 0; --i) {

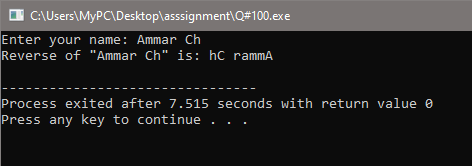
cout << userName[i];

}

cout << endl;

return 0;

}



**Q # 101:**

#include <iostream>

#include <cstring>

using namespace std;

int countCharacter( char str[], char ch) {

int count = 0;

int length = strlen(str);

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

if (str[i] == ch) {

count++;

}

}

return count;

}

int main() {

int maxSize = 100;

char userString[maxSize];

char searchChar;

cout << "Enter a string: ";

cin.getline(userString, maxSize);

cout << "Enter a character to search: ";

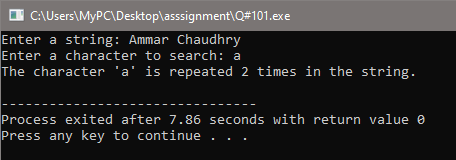
cin >> searchChar;

int charCount = countCharacter(userString, searchChar);

cout << "The character '" << searchChar << "' is repeated " << charCount << " times in the string." << endl;

return 0;

}



**Q # 102:**

#include <iostream>

using namespace std;

int main() {

const int rows = 2;

const int columns = 3;

int arr[rows][columns] = {{1, 2, 3}, {4, 5, 6}};

cout << "Values of the 2D array:\n";

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

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

cout << arr[i][j] << " ";

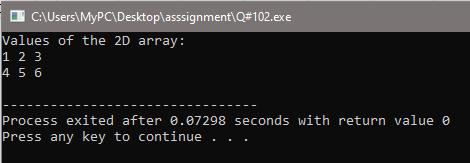
}

cout << endl;

}

return 0;

}



**Q # 103:**

#include <iostream>

using namespace std;

int main() {

const int rows = 4;

const int columns = 2;

int arr[rows][columns] = {{3, 8}, {1, 7}, {5, 12}, {9, 4}};

int minValue = arr[0][0];

int maxValue = arr[0][0];

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

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

if (arr[i][j] < minValue) {

minValue = arr[i][j];

}

if (arr[i][j] > maxValue) {

maxValue = arr[i][j];

}

}

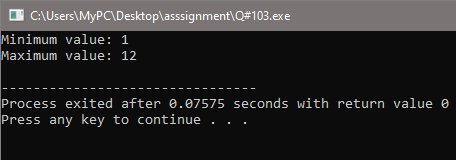
}

cout << "Minimum value: " << minValue << endl;

cout << "Maximum value: " << maxValue << endl;

return 0;

}



**Q # 104:**

#include <iostream>

using namespace std;

const int NUM\_CITIES = 2;

const int NUM\_DAYS = 7;

int main() {

int temperatures[NUM\_CITIES][NUM\_DAYS];

for (int city = 0; city < NUM\_CITIES; ++city) {

cout << "Enter temperatures for City " << city + 1 << " for the week:\n";

for (int day = 0; day < NUM\_DAYS; ++day) {

cout << "Day " << day + 1 << ": ";

cin >> temperatures[city][day];

}

}

cout << "\nTemperature data:\n";

for (int city = 0; city < NUM\_CITIES; ++city) {

cout << "City " << city + 1 << ":\n";

for (int day = 0; day < NUM\_DAYS; ++day) {

cout << "Day " << day + 1 << ": " << temperatures[city][day] << " degrees\n";

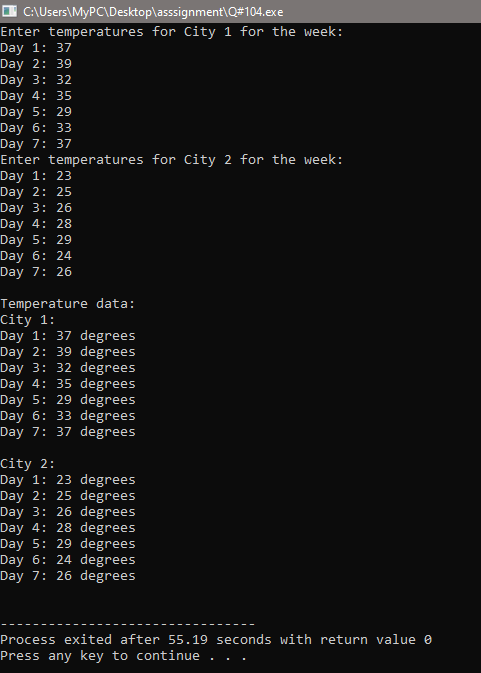
}

cout << endl;

}

return 0;

}



**Q # 105:**

#include <iostream>

using namespace std;

const int SIZE = 3;

void addMatrices(const int matrix1[][SIZE], const int matrix2[][SIZE], int result[][SIZE]) {

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

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

result[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

}

void displayMatrix(const int matrix[][SIZE]) {

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

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

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

int main() {

int matrix1[SIZE][SIZE], matrix2[SIZE][SIZE], resultMatrix[SIZE][SIZE];

cout << "Enter elements for the first matrix (3x3):\n";

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

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

cout << "Enter element at position (" << i + 1 << ", " << j + 1 << "): ";

cin >> matrix1[i][j];

}

}

cout << "Enter elements for the second matrix (3x3):\n";

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

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

cout << "Enter element at position (" << i + 1 << ", " << j + 1 << "): ";

cin >> matrix2[i][j];

}

}

addMatrices(matrix1, matrix2, resultMatrix);

cout << "\nResultant Matrix (Sum of the two matrices):\n";

displayMatrix(resultMatrix);

return 0;

}

