**1. Initialize an array of 10 elements and print the array elements both in normal and reverse order.**

**Answer:**

#include <iostream>

using namespace std;

int main() {

int arr[10] = {12, 32, 43, 1, 54, 53, 15, 64, 3, 13};

cout << "Showing the values:" << endl;

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

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

}

cout << endl;

cout << "Showing the reverse:" << endl;

for (int i = 9; i >= 0; i--) {

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

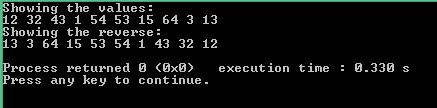
}

cout << endl;

return 0;

}

**Output:**



**2. Initialize an integer array of 10 elements and print how many numbers are odd and how many numbers are even.**

**Answer:**

#include <iostream>

using namespace std;

int main() {

int arr[10] = {12, 32, 43, 1, 54, 53, 15, 64, 3, 13};

int oddnum = 0;

int evennum = 0;

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

if (arr[i] % 2 == 0) {

evennum++;

} else {

oddnum++;

}

}

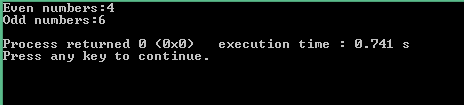
cout << "Even numbers:" <<evennum << endl;

cout << "Odd numbers:" <<oddnum << endl;

return 0;

}

**Output:**



**3.Write a function that takes TWO parameters to print all the odd numbers between a given range. Input the starting value of the range and ending value of the range. Then, send them as the parameters to your function.**

**Answer:**

#include <iostream>

using namespace std;

void pracTO(int a, int b) {

for (int i = a; i <= b; i++) {

if (i % 2 != 0) {

cout<<i<< " ";

}

}

cout<<endl;

}

int main() {

int start;

int ending;

cout<< "Enter the start number: ";

cin >> start;

cout<< "Enter the ending number: ";

cin >> ending;

cout<< "Given the start number: " << start << endl;

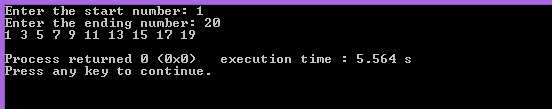
cout << "Given the ending number: " << ending << endl;

pracTO(start, ending);

return 0;

}

**Output:**



1. **Write a program to perform matrix addition between 3 matrices.**

**Answer:**

#include <iostream>

using namespace std;

int main(){

int m, n, c, d, first[10][10], second[10][10],third[10][10],sum[10][10];

cout << "Enter number of rows and columns of matrix:"<<" ";

cin >> m >> n;

cout << "Enter elements of first matrix:"<<" ";

for (c = 0; c < m; c++)

for (d = 0; d < n; d++)

cin >> first[c][d];

cout << "Enter the elements of second matrix:"<<" ";

for (c = 0; c < m; c++)

for (d = 0; d < n; d++)

cin >> second[c][d];

cout<<"Enter the elements of third matrix:"<<" ";

for (c = 0; c < m; c++)

for (d = 0; d < n; d++)

cin >> third[c][d];

for (c = 0; c < m; c++)

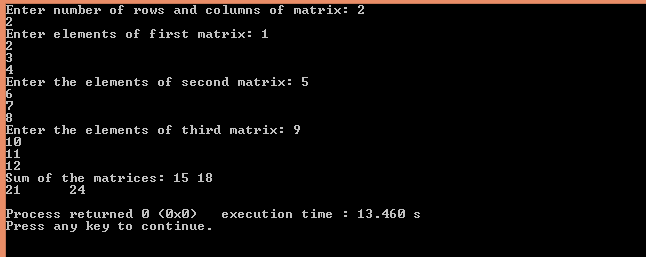
for (d = 0; d < n; d++)

sum[c][d] = first[c][d] + second[c][d]+third[c][d];

cout << "Sum of the matrices:"<<" ";

for (c = 0; c < m; c++) {

**Output:**



for (d = 0; d < n; d++)

cout << sum[c][d] << "\t";

cout << endl;

}

return 0;

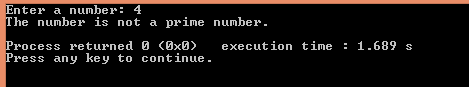
}

**5. Write a function to calculate factorial of a given integer number if that number is a prime number. If it is not, it will give an error.**

**Answer:**

#include <iostream>

**Output:**

****

using namespace std;

int isPrime(int n) {

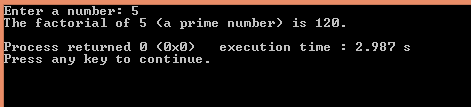
if (n <= 1) return 0;

for (int i = 2; i \* i <= n; ++i) {

if (n % i == 0) return 0;

}

return 1;



}

int factorial(int n) {

int result = 1;

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

result \*= i;

}

return result;

}

int main() {

int num;

cout << "Enter a number: ";

cin >> num;

if (isPrime(num)) {

cout << "The factorial of " << num << " (a prime number) is " << factorial(num) << "." << endl;

} else {

cout << "The number is not a prime number." << endl;

}

return 0;

}

**6. Initialize TWO integer arrays of different sizes. Merge the input arrays and create a new array. Then print the new array in reverse order.**

**Answer:**

#include <iostream>

using namespace std;

int main() {

int size1, size2;

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

cin >> size1;

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

cin >> size2;

int arr1[size1], arr2[size2];

cout << "Enter " << size1 << " integers for the first array: ";

for (int i = 0; i < size1; i++) {

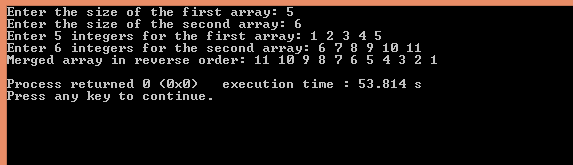
cin >> arr1[i];

}

cout << "Enter " << size2 << " integers for the second array: ";

for (int i = 0; i < size2; i++) {

**Output:**



cin >> arr2[i];

}

int mergedSize = size1 + size2;

int mergedArr[mergedSize];

for (int i = 0; i < size1; i++) {

mergedArr[i] = arr1[i];

}

for (int i = 0; i < size2; i++) {

mergedArr[size1 + i] = arr2[i];

}

cout << "Merged array in reverse order: ";

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

cout << mergedArr[i] << " ";

}

cout << endl;

return 0;

}

**7. Initialize TWO integer arrays A and B of different sizes. Make a new array with the common elements between A and B. Print the new array element(s). If there is no common element, output “No common element!” .**

**Answer:**

#include<iostream>

using namespace std;

int main() {

int n1, n2;

cout << "Enter your first array size: ";

cin >> n1;

int array1[n1];

cout << "Enter the elements of the first array: ";

for (int i = 0; i < n1; i++) {

cin >> array1[i];

}

cout << "\nEnter your second array size: ";

cin >> n2;

int array2[n2];

cout << "Enter the elements of the second array: ";

for (int i = 0; i < n2; i++) {

cin >> array2[i];

}

bool hasCommon = false;

cout << "\nYour common elements of the two arrays: ";

for (int i = 0; i < n1; i++) {

for (int j = 0; j < n2; j++) {

if (array1[i] == array2[j]) {

cout << array1[i] << " ";

hasCommon = true;

break;

}

}

}

if (!hasCommon) {

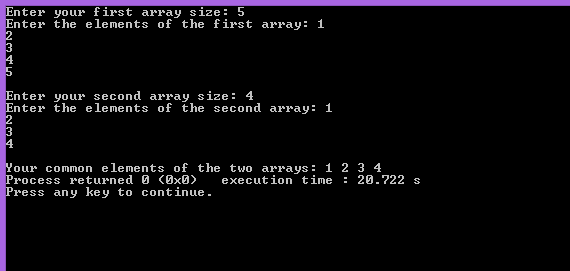
cout << "No common element";

}

return 0;

}

**Output:**



**8. Initialize an array. Size should be more than FIVE. Write you program to change the array in such a way so that there cannot be any duplicate element in the array anymore. Print the changed array. If the initialized array already had no duplicate elements from the beginning, output a message saying “Array already unique!”**

**Answer:**

#include <iostream>

using namespace std;

int main() {

int size;

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

cin >> size;

int arr[size];

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

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

cin >> arr[i];

}

int duplicateFound = 0;

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

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

if (arr[i] == arr[j]) {

arr[j] = -1;

duplicateFound = 1;

}

}

}

if (duplicateFound == 0) {

cout << "Array already unique!" << endl;

}

else{

cout << "Array after removing duplicates: ";

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

if (arr[i] != -1) {

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

}

}

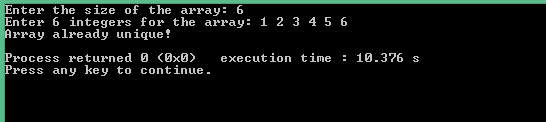
cout << endl;

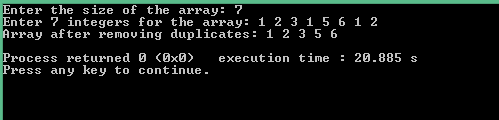
}

return 0;

}

**Output:**





**9. Initialize an integer array A of size 10. Take an integer as input and print how many times that integer occurs in A.**

**Answer:**

#include <iostream>

using namespace std;

int main() {

int A[10];

int num, count = 0;

cout << "Enter 10 integers for the array: " << endl;

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

cin >> A[i];

}

cout << "Input a number to search: ";

cin >> num;

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

if (A[i] == num) {

count++;

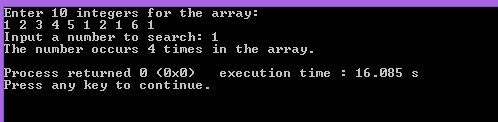
}

} cout << "The number occurs " << count << " times in the array." << endl;

return 0;

}

**Output:**

****

**10. Initialize an integer array of size 10. Print the number of time each element occurs in the array.**

**Answer:**

#include <iostream>

using namespace std;

int main() {

int A[10];

int count;

cout << "Enter 10 integers for the array: " << endl;

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

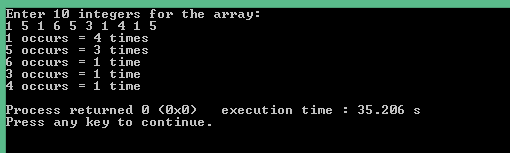
cin >> A[i];

}

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

count = 0;

**Output:**



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

if (A[i] == A[j]) {

count = -1;

break;

}

}

if (count != -1) {

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

if (A[i] == A[j]) {

count++;

}

}

if (count == 1) {

cout << A[i] << " occurs = " << count << " time" << endl;

} else {

cout << A[i] << " occurs = " << count << " times" << endl;

}

}

}

return 0;

}

**11.Initialize a matrix of minimum 3x4 (row x column) size. Output its transpose matrix.**

**Answer:**

#include <iostream>

using namespace std;

int main() {

int m, n;

int A[10][10];

int transpose[10][10];

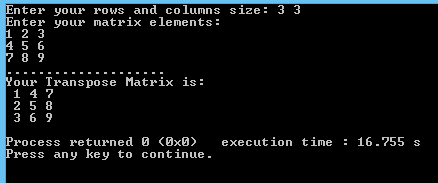
cout << "Enter your rows and columns size: ";

cin >> m >> n;

cout << "Enter your matrix elements:" << endl;

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

**Output:**



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

cin >> A[i][j];

}

}

cout << "...................." << endl;

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

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

transpose[j][i] = A[i][j];

}

}

cout << "Your Transpose Matrix is:" << endl;

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

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

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

}

cout << endl;

}

return 0;

}

12**. Write a program with appropriate data structure to keep records of 10 students. Each student will have the following information:**

1. Unique ID (you can use integer for this)
2. Number of Credits Completed
3. CGPA

**Print all the student’s ID whose CGPA is more than 3.75.**

**Print all the student’s ID who has completed more than 50 credits.**

**Answer:**

#include <iostream>

using namespace std;

struct student {

int id;

int credit;

float cgpa;

};

int main() {

student s[10];

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

cout << "Enter details for Student[" << i + 1 << "] (ID, Credit, CGPA):\n";

cin >> s[i].id >> s[i].credit >> s[i].cgpa;

if (s[i].credit < 0) {

cout << "Credit cannot be negative. Re-enter details for this student.\n";

i--;

continue;

}

if (s[i].cgpa < 0.0 || s[i].cgpa > 4.0) {

cout << "CGPA must be between 0.0 and 4.0. Re-enter details for this student.\n";

i--;

continue;

}

}

cout << "\nAll students' data:\n";

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

cout << "Student[" << i + 1 << "]\n";

cout << "ID: " << s[i].id << "\n";

cout << "Credits Completed: " << s[i].credit << "\n";

cout << "CGPA: " << s[i].cgpa << "\n";

cout << "-------------------------\n";

}

cout << "\nStudents with CGPA > 3.75:\n";

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

if (s[i].cgpa > 3.75) {

cout << "ID: " << s[i].id << " | CGPA: " << s[i].cgpa << "\n";

}

}

cout << "\nStudents who completed more than 50 credits:\n";

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

if (s[i].credit > 50) {

cout << "ID: " << s[i].id << " | Credits: " << s[i].credit << "\n";

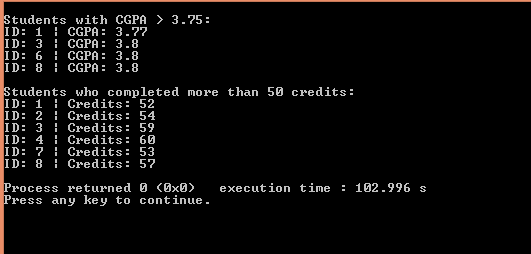
}

}

return 0;

}

**Output:**



**Linear Searching**

**Answer:**

#include <iostream>

using namespace std;

int main() {

int size, key;

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

cin >> size;

int arr[size];

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

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

cin >> arr[i];

}

cout << "Enter the number to search: ";

cin >> key;

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

if (arr[i] == key) {

cout << "Element found at index " << i << endl;

return 0;

}

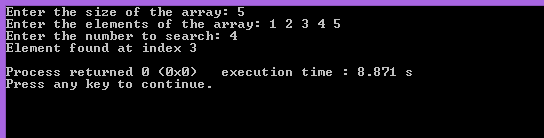
}

cout << "Element not found." << endl;

return 0;

}

**Output:**



**1.Write a C++ code to implement Bubble Sort**

**Answer:**

#include <iostream>

using namespace std;

int main() {

int n;

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

cin >> n;

int arr[n];

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

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

cin >> arr[i];

}

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

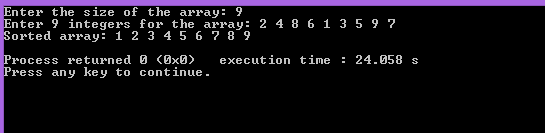
for (int j = 0; j < n - i - 1; j++) {

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

swap(arr[j], arr[j + 1]);

}

**Output:**



}

}

cout << "Sorted array: ";

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

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

}

cout << endl;

return 0;

}