PROGRAMMING FUNDAMENTALS

Let's explore technology
together to live in the future

Checkout more on
https://github.com/Sy-hash-collab

Sy-hash-collab

C ++ PAST PAPERS PRORGRAMS

```
#include<iostream>
using namespace std;
int main()
{
int n=0;
cout<<"Enter An Integer\n";
cin>>n;
int choice=0;
cout<<"Enter 1 For + to - Conversion\nEnter 2 for - to + Conversion\n";
cin>>choice;
switch(choice)
  {
    case 1:
      {
      if ( n > 0 )
      cout << " Converted integer is " << -1 * n
      else cout << " This Is Not a Positive I
      break ;
    }
  case 2:
    {
    if ( n < 0 )
    cout << " Converted integer is " << -1 * n ;
    else cout << " This Is Not a Negative I
    break ;
```

```
  }       }
  r  e  t
  u  r  n
  0  ;
}  #include<iostream>
using namespace std;
int main() {



   int salary=0,HRA=0,DA=0,gross=0;
   cout<<"Enter Salary\n";
   cin>>salary;
   if(salary<=10000)
{
   HRA=salary*0.2;
 DA=salary*0.8;
gross=salary+HRA+DA;
cout<<"Gross Salary Is ="<<gross;


}
else if (salary<=20000)
{
   HRA=salary *0. 25;
 DA=salary*0.9;
gross=salary+HRA+DA;
cout<<"Gross Salary Is ="<<gross;
}
```

```
else if (salary>20000)
{
HRA=salary*0.3;
DA=salary*0.95;
gross=salary+HRA+DA;
cout<<"Gross Salary Is ="<<gross;
}
cout<<"\nProgram Ends!!";
return 0;
}
```

Q # 3

```
#include <iostream>
using namespace std;
// Summ of all even and Odd Numbers in an array of 1 to 8.
int main()
{
   int arr[]={1,2,3,4,5,6,7,8};
 int osum=0,esum=0;
   for (int i = 0; i < 8; i++)
   {
      cout<<arr[i]<<" ";
   }
   f o r   ( i n t   i   =   0 ;   i   <   8 ;   i + + )
   {
      if (arr[i]%2==0)
      {
```

```
        esum =esum + arr[i];
      }
      i f    ( a r r [ i ] % 2 ! = 0 )
      {
        osum =osum + arr[i];
      }
   }
```

cout<<"\nThe Sum is of Even Integers is: "<<esum; cout<<"\nThe Sum is of Odd Integers is: "<<osum; cout<<endl<<"Program Ends!"; return 0; } Q # 4 #include <iostream> using namespace std; //Calcluate %age and Grade of marks input by user of 5 Subjects. // % >= 90%: Grade A // % >= 90%: Grade B // % >= 90%: Grade C // % >= 90%: Grade D // % >= 90%: Grade E // % >= 90%: Grade F int main() {

```
   float p;
   int t=500;
char G;
```

```
 int mmaths;
cout<<"\nEnter The Marks of Maths out of 100: ";
cin>>mmaths;
int mbio;
cout<<"\nEnter The Marks of Biology out of 100: ";
cin>>mbio;
int mchem;
cout<<"\nEnter The Marks of Chemistry out of 100: ";
cin>>mchem;
int meng;
cout<<"\nEnter The Marks of English out of 100: ";
cin>>meng;
int murdu;
cout<<"\nEnter The Marks of Urdu out of 100: ";
cin>>murdu;
p=(mmaths+mbio+mchem+meng+murdu)/5;

  cout<<"\nThe Percentage is: "<<p;
  if (p>=90)
  {
    cout<<"\nGrade A";
  }
  e l s e   i f   ( p > = 8 0   & &   p < 9 0 )
  {
    cout<<"\nGrade B";
  }
  e l s e   i f   ( p > = 7 0   & &   p < 8 0 )
```

```
  {
    cout<<"\nGrade C";
  }
  else if (p >= 60 && p < 70)
  {
    cout<<"\nGrade D";
  }
  else if (p >= 50 && p < 60)
  {
    cout<<"\nGrade E";
  }
  else if (p >= 40 )
  {
    cout<<"\nGrade F";
  }
cout<<endl<<"Program Ends!";
return 0;
}
```

Q # 5

write a program in c++ to calculate percentage and display grade of five students of five subjects each by inputting marks from user the marks for each subject are 20 .

```
#include <iostream>
#include <iomanip>
using namespace std;

// Function to calculate the percentage
double calculatePercentage(int marks[], int totalSubjects) {
```

```
 int totalMarks = totalSubjects * 20; // Assuming each subject has a maximum of 20 marks
int obtainedMarks = 0;

    for (int i = 0; i < totalSubjects; i++) {
        obtainedMarks += marks[i];
    }

    return (static_cast<double>(obtainedMarks) / totalMarks) * 100.0;
}

// Function to determine the grade based on the percentage
char determineGrade(double percentage) {
    if (percentage >= 90.0) {
        return 'A';
    } else if (percentage >= 80.0) {
        return 'B';
    } else if (percentage >= 70.0) {
        return 'C';
    } else if (percentage >= 60.0) {
        return 'D';
    } else if (percentage >= 40.0) {
        return 'E';
    } else {
        return 'F';
    }
}

int main() {
```

```
 const int numStudents = 5;
const int numSubjects = 5;


   int marks[numStudents][numSubjects];


   // Input marks for each student
 for (int i = 0; i < numStudents; i++) {
       cout << "Enter marks for student " << (i + 1) << ":" << endl;


       for (int j = 0; j < numSubjects; j++) {
          cout << "Enter marks for subject " << (j + 1) << ": ";
          cin >> marks[i][j];
       }


       cout << endl;
   }


   // Calculate percentage and display grade for each student
 for (int i = 0; i < numStudents; i++) {
       double percentage = calculatePercentage(marks[i], numSubjects);
       char grade = determineGrade(percentage);


       cout << "Student " << (i + 1) << ": ";
       cout << "Percentage: " << fixed << setprecision(2) << percentage << "%, ";
       cout << "Grade: " << grade << endl;
   }


   return 0;
```

```
}
```
Q # 5

```
#include <iostream>

using namespace std;


int main()

{

   int a;

 cout<<"Enter The Rows: ";

   cin>>a;

   for (int i = 0; i < a; i++)

   {

      for (int j = 1; j < a-i; j++)

      {

         cout<<j+i;

      }

      c   o   u   t   <   <   e   n   d   l   ;

   }


cout<<endl<<"Program  Ends!";

return 0; }
```
Q # 6 #include <iostream>

```
using namespace std;
```

```
// Factorial
void fact(int&);

int main()
{
    int num = 0;
    cout << "Enter a Number: ";
    cin >> num;
    fact(num);
    cout << endl << "Program Ends!";
    return 0;
}

void fact(int& n)
{
    long int f = 1;
    for (int i = 1; i <= n; i++)
    {
        f = f * i;
    }
    cout << "Factorial of " << n << " is: " << f;
}
```

Q # 7

```
#include <iostream>
using namespace std;
//s = student name, rn, marks F
// display detail of student
struct student
```

```
{
    char name[50];
    int rollnumber;
    float marks[5];
};

int main()
{
    student s1;
    cout<<"Enter Name: ";
    cin.getline(s1.name,45);
    cout<<"\nEnter Roll Number: ";
    cin>>s1.rollnumber;
    cout<<"\nEnter Marks of 5 Subjects:- \n";
    for (int i = 0; i < 5; i++)
    {
        cout<<"Enter The Marks of Subject 0"<<i+1<<" :";
        cin>>s1.marks[i];
    }
    cout<<"The Detail of Student is:-\n";
    cout<<s1.name<<endl<<s1.rollnumber<<endl;
    for (int i = 0; i < 5; i++)
    {
        cout<<"The Marks of Subject 0"<<i+1<<" are: ";
        cout<<s1.marks[i]<<endl;
    }
    i  n  t     t  t  =  0  ;
    for (int i = 0; i < 5; i++)
```

```
    {
        tt=tt+s1.marks[i];
    }
    cout<<"\nThe Total Makrs are: "<<tt;

cout<<endl<<"Program Ends!";
return 0;
}
```

Q # 8

```
// BS-CSEC-1A
// Question 03
// input an array with dimentions

#include <iostream>
using namespace std;

int main()
{
    int arr[4][1][5][3];
    cout<<"Enter the elements in array of dimentions [4][1][5][3]:"<<endl;
    for (int i = 1; i <= 4; i++)
    {
        for (int j = 1; j <= 1; j++)
        {
            for (int k = 1; k <= 5; k++)
            {
                for (int l = 1; l <= 3; l++)
                {
```

```
                cin>>arr[i][j][k][l];
            }
        }
    }
}
cout<<"\nYour inputted array is:"<<endl;
for (int i = 1; i <= 4; i++)
{
    for (int j = 1; j <= 1; j++)
    {
        for (int k = 1; k <= 5; k++)
        {
            for (int l = 1; l <= 3; l++)
            {
                cout<<arr[i][j][k][l]<<" ";
            }
            cout<<endl;
        }
        cout<<endl;
    }
    cout<<endl;
}
return 0;
}
```

Q # 9

/ BS-CSEC-1A

// Question 04

// array of 10 digits sorts the array using bubble sort and search using binary search

```
#include <iostream>
using namespace std;

int main()
{
    int arr[10];
    cout<<"Enter the elements in 10 digit array:"<<endl;
    for (int i = 0; i < 10; i++)
    {
        cin>>arr[i];
    }
    cout<<"\nThe array sorted using bubble sort is: ";
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            if (arr[j]>arr[j+1])
            {
                int temp=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=temp;
            }
        }
    }
    for (int i = 0; i < 10; i++)
    {
        cout<<arr[i]<<" ";
```

```
    }
    i n t    k e y = 0 ;
    cout<<"\nEnter the element to search using binary search = ";
    cin>>key;
    int mid=0,s=0,e=10,loc=-1;
    while (s<=e)
    {
        mid=(s+e)/2;
        if (arr[mid]==key)
        {
            loc=mid;
            cout<<"The element found at index "<<loc<<endl;
            break;
        }
        e l s e   i f   ( a r r [ m i d ] > k e y )
        {
            e=mid-1;
        }
        else if (arr[mid]<key)
        {
            s=mid+1;
        }
    }
    return 0;
}
```

Q # 10

// BS-CSEC-1A

// Question 05

// enter the number till the user wants and at the end dispaly count of positive , negative and zeros entered

```
#include <iostream>

using namespace std;


int main()

{

    int x=0,posnum=0,negnum=0,zeros=0;

    for ( ; ; x++)

    {

        cout<<"\nEnter the number = ";

        cin>>x;

        if (x>0)

        {

            posnum++;

        }

        if (x<0)

        {

            negnum++;

        }

        i  f      (  x  =  =  0  )

        {

            zeros++;

        }

        cout<<"To exit enter -1 \nOR\n";

        if (x==-1)

        {
```

```
            break;
        }
    }
    cout<<"Positive numbers entered = "<<posnum<<endl;
    cout<<"Negative numbers entered = "<<negnum<<endl;
    cout<<"Zeros entered = "<<zeros<<endl;
    return 0;
}
Q # 11
// question no.02
// cyber security

#include <iostream>
using namespace std;

struct carinfo
{
    int carnum;
    int ownercnic;
    char carmodel[30];
    int kmsrun;
}c[5];
int main()
{
    cout<<"Enter the information of 5 cars:"<<endl;
    for (int i = 1; i <= 5; i++)
    {
        cout<<"Enter the information of car "<<i<<endl;
```

```
        cout<<"Enter the car number = ";
        cin>>c[i].carnum;
        cout<<"Enter the owner cnic = ";
        cin>>c[i].ownercnic;
        cout<<"Enter the car model = ";
        cin.ignore();
        cin.getline(c[i].carmodel,30);
        cout<<"Enter the kms run = ";
        cin>>c[i].kmsrun;
        cout<<endl;
    }
    cout<<"You entered the following information:"<<endl;
    for (int i = 1; i <= 5; i++)
    {
        cout<<"\nThe information of car "<<i<<endl;
        cout<<"\nThe car number = ";
        cout<<c[i].carnum<<endl;
        cout<<"The owner cnic = ";
        cout<<c[i].ownercnic<<endl;
        cout<<"The car model = ";
        cout<<c[i].carmodel<<endl;
        cout<<"The kms run = ";
        cout<<c[i].kmsrun<<endl;
        cout<<endl;
    }
    return 0;
}
```

Q # 12

Write a structure to store the name , account number and balance of 50 customers and store their information

1 . Write a function to print the names of all customers having balance less than $200

2 . Write a function to add $100 in the balance of all customers having more than $1000 in their balance and then print the incremented value of their balance

```
#include <iostream>
#include <string>
using namespace std;

// Structure to store customer information
struct Customer {
    string name;
    int accountNumber;
    double balance;
};

// Function to print names of customers with balance less than $200
void printCustomersWithLowBalance(const Customer customers[], int size) {
    cout << "Customers with balance less than $200:" << endl;

    for (int i = 0; i < size; i++) {
        if (customers[i].balance < 200.0) {
            cout << customers[i].name << endl;
        }
    }

    cout << endl;
}
```

```
// Function to add $100 to balance of customers with more than $1000
void addBalanceForRichCustomers(Customer customers[], int size) {
    cout << "Incremented balances of customers with more than $1000:" << endl;

    for (int i = 0; i < size; i++) {
        if (customers[i].balance > 1000.0) {
            customers[i].balance += 100.0;
            cout << customers[i].name << ": $" << customers[i].balance << endl;
        }
    }

    cout << endl;
}

int main() {
    const int numCustomers = 50;
    Customer customers[numCustomers];

    // Input customer information
    for (int i = 0; i < numCustomers; i++) {
        cout << "Enter details for customer " << (i + 1) << ":" << endl;

        cout << "Name: ";
        cin >> customers[i].name;

        cout << "Account Number: ";
        cin >> customers[i].accountNumber;
```

```
        cout << "Balance: $";
        cin >> customers[i].balance;

        cout << endl;
    }

    // Print names of customers with balance less than $200
 printCustomersWithLowBalance(customers, numCustomers);

    // Add $100 to balance of customers with more than $1000 and print incremented balances
    addBalanceForRichCustomers(customers, numCustomers);

    return 0;
}
```

Q # 13

1. **Student Database**:

```
cppCopy code
#include <iostream>
#include <string>
using namespace std;

struct Student {
    string name;
    int rollNumber;
    int age;
};
```

```
int main() {
    const int numStudents = 3;
    Student students[numStudents];

    // Input student information
    for (int i = 0; i < numStudents; i++) {
        cout << "Enter details for student " << (i + 1) << ":" << endl;

        cout << "Name: ";
        cin >> students[i].name;

        cout << "Roll Number: ";
        cin >> students[i].rollNumber;

        cout << "Age: ";
        cin >> students[i].age;

        cout << endl;
    }

    // Display student information
    cout << "Student Information:" << endl;
    for (int i = 0; i < numStudents; i++) {
        cout << "Name: " << students[i].name << endl;
        cout << "Roll Number: " << students[i].rollNumber << endl;
        cout << "Age: " << students[i].age << endl;
        cout << endl;
    }
```

```
    return 0;
}
```

This program creates a student database using a structure. It allows the user to input the name, roll number, and age of each student. It then displays the information for all the students.

1. **Book Inventory**:

cppCopy code

```
#include <iostream>
#include <string>
using namespace std;

struct Book {
    string title;
    string author;
    int year;
};

int main() {
    const int numBooks = 3;
    Book books[numBooks];

    // Input book information
    for (int i = 0; i < numBooks; i++) {
        cout << "Enter details for book " << (i + 1) << ":" << endl;

        cout << "Title: ";
        cin.ignore();
        getline(cin, books[i].title);
```

```
        cout << "Author: ";
        getline(cin, books[i].author);

        cout << "Year: ";
        cin >> books[i].year;

        cout << endl;
    }

    // Display book information
    cout << "Book Inventory:" << endl;
    for (int i = 0; i < numBooks; i++) {
        cout << "Title: " << books[i].title << endl;
        cout << "Author: " << books[i].author << endl;
        cout << "Year: " << books[i].year << endl;
        cout << endl;
    }

    return 0;
}
```

This program creates a book inventory using a structure. It allows the user to input the title, author, and year of each book. It then displays the information for all the books.

Program 3: Bank Account Management

cppCopy code

```
#include <iostream>
#include <string>
using namespace std;
```

```
struct BankAccount {
    int accountNumber;
    string accountHolderName;
    double balance;
};

void deposit(BankAccount& account, double amount) {
    account.balance += amount;
}

void withdraw(BankAccount& account, double amount) {
    if (account.balance >= amount) {
        account.balance -= amount;
    } else {
        cout << "Insufficient balance!" << endl;
    }
}

void checkBalance(const BankAccount& account) {
    cout << "Account Number: " << account.accountNumber << endl;
    cout << "Account Holder Name: " << account.accountHolderName << endl;
    cout << "Balance: $" << account.balance << endl;
}

int main() {
    const int numAccounts = 3;
    BankAccount accounts[numAccounts];
```

```
for (int i = 0; i < numAccounts; i++) {
    cout << "Enter details for account " << (i + 1) << ":" << endl;

    cout << "Account Number: ";
    cin >> accounts[i].accountNumber;

    cout << "Account Holder Name: ";
    cin.ignore();
    getline(cin, accounts[i].accountHolderName);

    cout << "Initial Balance: $";
    cin >> accounts[i].balance;

    cout << endl;
}

for (int i = 0; i < numAccounts; i++) {
    cout << "Enter deposit amount for account " << (i + 1) << ": $";
    double depositAmount;
    cin >> depositAmount;
    deposit(accounts[i], depositAmount);
}

for (int i = 0; i < numAccounts; i++) {
    cout << "Enter withdrawal amount for account " << (i + 1) << ": $";
    double withdrawalAmount;
    cin >> withdrawalAmount;
```

```
        withdraw(accounts[i], withdrawalAmount);
    }

    for (int i = 0; i < numAccounts; i++) {
        cout << "Details for account " << (i + 1) << ":" << endl;
        checkBalance(accounts[i]);
        cout << endl;
    }

    return 0;
}
```

Program 4: Employee Payroll

cppCopy code

```
#include <iostream>
#include <string>
using namespace std;

struct Employee {
string name;
int employeeID;
    double monthlySalary;
};

 void calculateAnnualSalary(const Employee& employee) {
     double annualSalary = employee.monthlySalary * 12;
    cout << "Employee Name: " << employee.name << endl;
    cout << "Employee ID: " << employee.employeeID << endl;
    cout << "Annual Salary: $" << annualSalary << endl;
```

```
}

int main() {
    const int numEmployees = 3;
    Employee employees[numEmployees];

    for (int i = 0; i < numEmployees; i++) {
        cout << "Enter details for employee " << (i + 1) << ":" << endl;

        cout << "Name: ";
        cin >> employees[i].name;

        cout << "Employee ID: ";
        cin >> employees[i].employeeID;

        cout << "Monthly Salary: $";
        cin >> employees[i].monthlySalary;

        cout << endl;
    }

    double thresholdSalary;
    cout << "Enter the threshold annual salary: $";
    cin >> thresholdSalary;

    for (int i = 0; i < numEmployees; i++) {
        calculateAnnualSalary(employees[i]);
        if (employees[i].monthlySalary * 12 > thresholdSalary) {
```

```
            cout << "Annual Salary exceeds threshold." << endl;
        }
        cout << endl;
    }

    return 0;
}
```

Program 5: Product Inventory

cppCopy code

```
#include <iostream>
#include <string>
using namespace std;

struct Product {
    string name;
    double price;
    int quantity;
};

void searchProductByName(const Product products[], int numProducts, const string& name) {
    cout << "Search Results for Product " << name << ":" << endl;
    for (int i = 0; i < numProducts; i++) {
        if (products[i].name == name) {
            cout << "Name: " << products[i].name << endl;
            cout << "Price: $" << products[i].price << endl;
            cout << "Quantity: " << products[i].quantity << endl;
        }
    }
```

```
    cout << endl;
}

void displayInventory(const Product products[], int numProducts) {
    cout << "Product Inventory:" << endl;
    for (int i = 0; i < numProducts; i++) {
        cout << "Name: " << products[i].name << endl;
        cout << "Price: $" << products[i].price << endl;
        cout << "Quantity: " << products[i].quantity << endl;
        cout << endl;
    }
}

int main() {
    const int numProducts = 3;
    Product products[numProducts];

    for (int i = 0; i < numProducts; i++) {
        cout << "Enter details for product " << (i + 1) << ":" << endl;

        cout << "Name: ";
        cin.ignore();
        getline(cin, products[i].name);

        cout << "Price: $";
        cin >> products[i].price;

        cout << "Quantity: ";
```

```
        cin >> products[i].quantity;

        cout << endl;
    }

    string productToSearch;
    cout << "Enter product name to search: ";
    cin.ignore();
    getline(cin, productToSearch);

    searchProductByName(products, numProducts, productToSearch);

    displayInventory(products, numProducts);

    return 0;
}
```

Q # 15

Write a program that inputs 3d array

1. Write a function that sorts the array row wise using bubble sort

2. Write a function that sorts the array column wise using bubble sort

3. Write a function that searches the array for any item entered by user using binary search

```
#include <iostream>
using namespace std;

const int ROWS = 3;
const int COLS = 3;
const int DEPTH = 3;
```

```
void printArray(int arr[ROWS][COLS][DEPTH]) {
 for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            for (int k = 0; k < DEPTH; k++) {
                cout << arr[i][j][k] << " ";
            }
            cout << endl;
        }
        cout << endl;
    }
}

void bubbleSortRow(int arr[ROWS][COLS][DEPTH], int row) {
    for (int i = 0; i < COLS * DEPTH - 1; i++) {
        for (int j = 0; j < COLS * DEPTH - i - 1; j++) {
            if (arr[row][j / DEPTH][j % DEPTH] > arr[row][(j + 1) / DEPTH][(j + 1) % DEPTH]) {
                int temp = arr[row][j / DEPTH][j % DEPTH];
                arr[row][j / DEPTH][j % DEPTH] = arr[row][(j + 1) / DEPTH][(j + 1) % DEPTH];
                arr[row][(j + 1) / DEPTH][(j + 1) % DEPTH] = temp;
            }
        }
    }
}

void sortRowWise(int arr[ROWS][COLS][DEPTH]) {
 for (int i = 0; i < ROWS; i++) {
        bubbleSortRow(arr, i);
```

```
    }
}

void bubbleSortColumn(int arr[ROWS][COLS][DEPTH], int col) {
 for (int i = 0; i < ROWS * DEPTH - 1; i++) {
        for (int j = 0; j < ROWS * DEPTH - i - 1; j++) {
            if (arr[j / DEPTH][col][j % DEPTH] > arr[(j + 1) / DEPTH][col][(j + 1) % DEPTH]) {
                int temp = arr[j / DEPTH][col][j % DEPTH];
                arr[j / DEPTH][col][j % DEPTH] = arr[(j + 1) / DEPTH][col][(j + 1) % DEPTH];
                arr[(j + 1) / DEPTH][col][(j + 1) % DEPTH] = temp;
            }
        }
    }
}

void sortColumnWise(int arr[ROWS][COLS][DEPTH]) {
    for (int i = 0; i < COLS; i++) {
        bubbleSortColumn(arr, i);
    }
}

bool binarySearch(int arr[ROWS][COLS][DEPTH], int item) {
    int low = 0;
 int high = ROWS * COLS * DEPTH - 1;

    while (low <= high) {
        int mid = low + (high - low) / 2;
        int value = arr[mid / (COLS * DEPTH)][(mid / DEPTH) % COLS][mid % DEPTH];
```

```
        if (value == item) {
            return true;
        } else if (value < item) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }

    return false;
}

int main() {
 int arr[ROWS][COLS][DEPTH];

    cout << "Enter elements of the 3D array:" << endl;
 for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            for (int k = 0; k < DEPTH; k++) {
                cin >> arr[i][j][k];
            }
        }
    }

    cout << endl << "Array before sorting:" << endl;
    printArray(arr);
```

```
 sortRowWise(arr);
    cout << endl << "Array after sorting row-wise:" << endl;
    printArray(arr);

 sortColumnWise(arr);
    cout << endl << "Array after sorting column-wise:" << endl;
    printArray(arr);

    int item;
    cout << endl << "Enter an item to search in the array: ";
    cin >> item;
    bool found = binarySearch(arr, item);
    if (found) {
        cout << "Item found in the array." << endl;
    } else {
        cout << "Item not found in the array." << endl;
    }

    return 0;
}
```