Task 1

Code:

#include <iostream>

using namespace std;

enum cousine{Italian=1,Arabic,Desi};

struct chef{

    string name;

    cousine specialisation;

};

struct order{

    int orderNum;

    cousine type;

    string name;

    string chef;

};

void addChef(int &totalChefs,chef \*&chefs){

    chef \*newChefs= new chef[totalChefs+1];

    int specialisationNum;

    for(int i=0;i<totalChefs;i++){

        newChefs[i]=chefs[i];

    }

    cout<<"\nEnter Chef Name : ";

    cin>>newChefs[totalChefs].name;

    cout<<"--Choose Chef Specialisation---";

    cout<<"\n1.Italian";

    cout<<"\n2.Arabic";

    cout<<"\n3.Desi";

    cout<<"\nSpecialisation : ";

    cin>>specialisationNum;

    newChefs[totalChefs].specialisation=static\_cast<cousine>(specialisationNum);

    cout<<"\n\033[32mChef Added to List\033[0m";

    chefs=newChefs;

    totalChefs++;

}

void addOrder(int &totalOrders,order \*&orders){

    int orderNum;

    order \*newOrders= new order[totalOrders+1];

    for(int i=0;i<totalOrders;i++){

        newOrders[i]=orders[i];

    }

    cout<<"\nEnter Dish Name : ";

    cin>>newOrders[totalOrders].name;

    cout<<"\n---Select Dish Type---";

    cout<<"\n1.Italian";

    cout<<"\n2.Arabic";

    cout<<"\n3.Desi";

    cout<<"\nType : ";

    cin>>orderNum;

    newOrders[totalOrders].type=static\_cast<cousine>(orderNum);

    newOrders[totalOrders].orderNum = totalOrders+1;

    cout<<"\n\033[32mOrder # "<<newOrders[totalOrders].orderNum<<" Submitted!\033[0m";

    orders=newOrders;

    totalOrders++;

}

void assignOrders(int &totalOrders,order \*&orders,int &totalChefs,chef \*&chefs){

    for(int i=0 ;i<totalOrders;i++){

        bool assigned=false;

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

            if(orders[i].type==chefs[j].specialisation){

                orders[i].chef=chefs[j].name;

                assigned=true;

                break;

            }

        }

        if(assigned==false){

            orders[i].chef="Chef of Speciality not available";

        }

    }

}

int main(){

    int totalChefs=0;

    chef \*chefs;

    int totalOrders=0;

    order \*orders;

    int newOrders,newChefs;

    void (\*listChef)(int&,chef\*&);

    void (\*listOrder)(int&,order\*&);

    void (\*assignment)(int&,order\*&,int&,chef\*&);

    listChef=addChef;

    listOrder=addOrder;

    assignment=assignOrders;

    cout<<"How many chefs need to be added : ";

    cin>>newChefs;

    for(int i=0;i<newChefs;i++){

        listChef(totalChefs,chefs);

    }

    cout<<"\nHow many Orders need to be added : ";

    cin>>newOrders;

    for(int i=0;i<newOrders;i++){

        listOrder(totalOrders,orders);

    }

    assignment(totalOrders,orders,totalChefs,chefs);

    for(int i=0;i<totalOrders;i++){

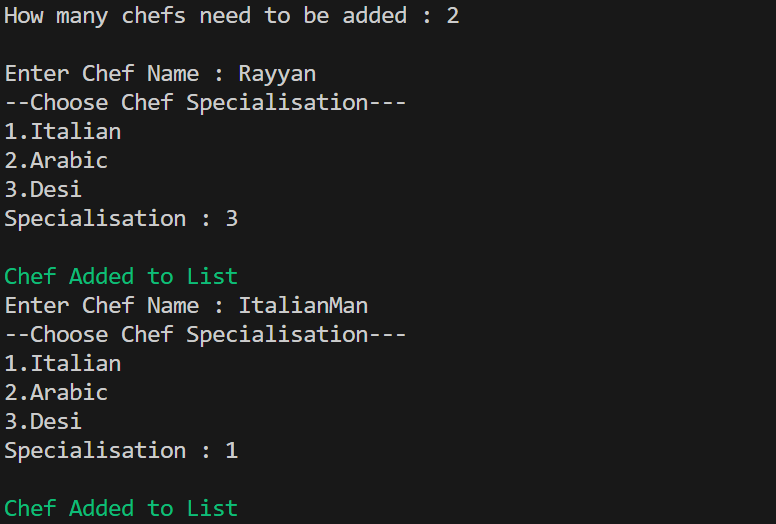
        cout<<"\nOrder #"<<orders[i].orderNum<<" has been assigned to "<<orders[i].chef;

    }

    return 0;

}

Images :



A screenshot of a computer program

Description automatically generated

A black background with white text

Description automatically generated

Task-2A

Code:

#include <iostream>

#include <conio.h>

using namespace std;

struct Employee{

    string name;

    int ID;

    string department;

    float salary;

};

void addEmployee(int &total,Employee \*&emp,int i){

    Employee \*newEmp= new Employee[total+1];

    for(int i=0;i<total;i++){

        newEmp[i]=emp[i];

    }

    cout<<"Enter Employee "<<i+1<<" Details\n";

    cout<<"Enter Employee Name : ";

    cin>>newEmp[total].name;

    cout<<"Enter Employee ID : ";

    cin>>newEmp[total].ID;

    cout<<"Enter Employee Department : ";

    cin>>newEmp[total].department;

    cout<<"Enter Employee Salary : ";

    cin>>newEmp[total].salary;

    delete[] emp;

    emp=newEmp;

    total++;

}

void displayEmployee(int searchID,int total,Employee \*emp){

    bool present =false;

    for(int i=0;i<total;i++){

        if (emp[i].ID==searchID){

            cout<<"\033[32mEmployee "<<i+1<<" Details";

            cout<<"\nName : "<<emp[i].name;

            cout<<"\nDepartment : "<<emp[i].department;

            cout<<"\nSalary : "<<emp[i].salary<<"\033[0m";

            present=true;

            break;

        }

    }

    if(present==false){

            cout<<"\033[31mNo such Employee was found\033[0m";

    }

}

float calcTotal(int total,Employee \*emp){

    float sum=0;

    for(int i=0;i<total;i++){

        sum+=emp[i].salary;

    }

    return sum;

}

int main(){

    Employee \*emp;

    int numEmployees=0,addNum,searchID,sNum;

    void (\*searchDetails)(int,int,Employee\*);

    float (\*totalSalary)(int,Employee\*);

    totalSalary=calcTotal;

    searchDetails=displayEmployee;

    system("cls");

    cout<<"How Many Employees Would you like to add : ";

    cin>>addNum;

    for(int i=0;i<addNum;i++){

        addEmployee(numEmployees,emp,i);

        system("cls");

    }

    cout<<"How many employees do you want to search: ";

    cin>>sNum;

    for(int i=0;i<sNum;i++){

        system("cls");

        cout<<"Enter ID of Employee You want : ";

        cin>>searchID;

        searchDetails(searchID,numEmployees,emp);

        cout<<"\nPress Any Key to continue";

        getch();

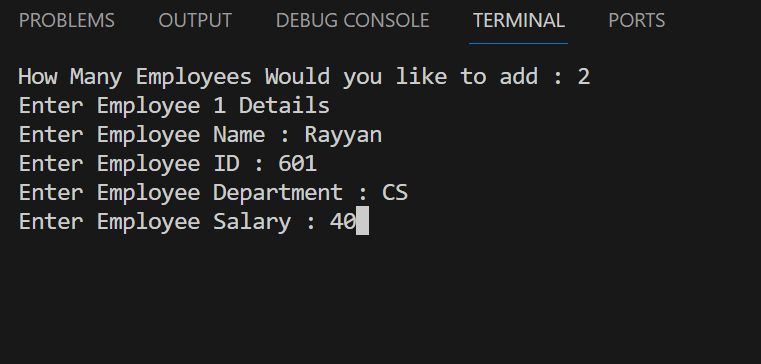
    }

    system("cls");

    cout<<"Sum of all Salaries : "<<totalSalary(numEmployees,emp);

    return 0;

}

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screen shot of a computer

Description automatically generated

Task-2B

Code:

#include <iostream>

using namespace std;

union EmployeeDetails{

    float salary;

    const char\* department;

};

int main(){

EmployeeDetails john;

john.salary=42000;

john.department="Marketing";

cout<<"-----Details-----";

cout<<"\nSalary : "<<john.salary;

cout<<"\nDepartment : "<<john.department;

cout<<"\n\nThe Union only stores the department as it only stores one data\nonly one memory slot to be used/shared by all members at once this can be\navoided by displaying after storing and repeating.\n";

john.salary=42000;

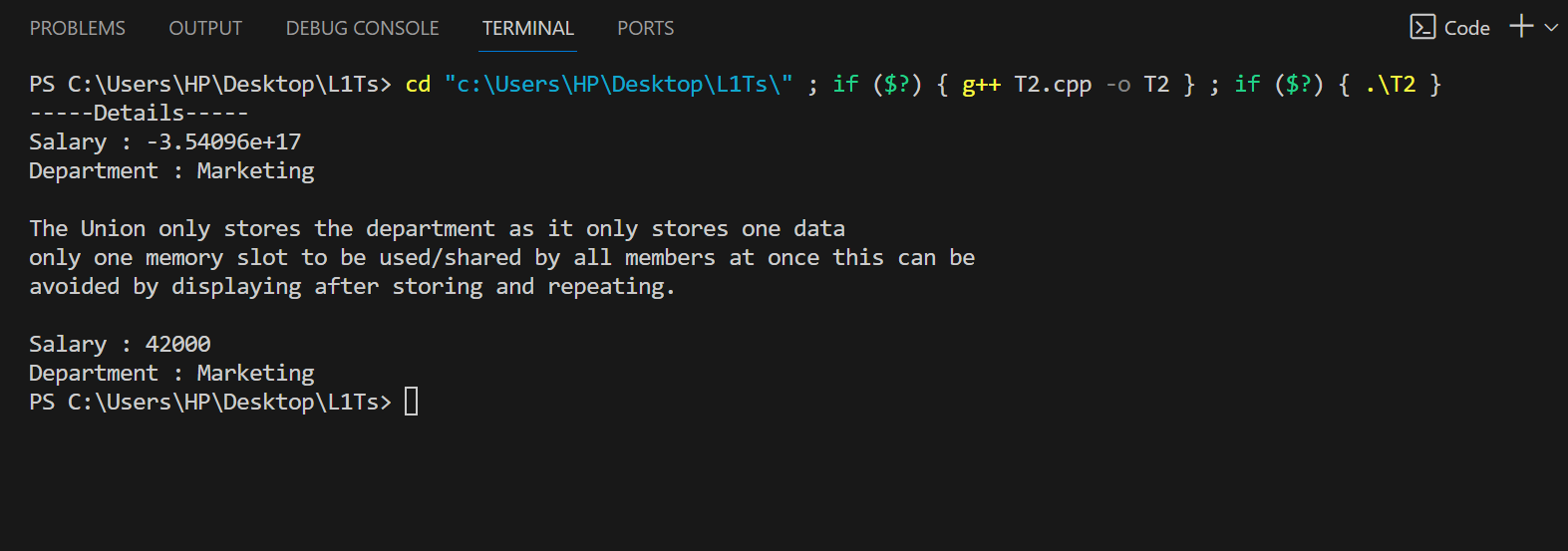
cout<<"\nSalary : "<<john.salary;

john.department="Marketing";

cout<<"\nDepartment : "<<john.department;

return 0;

}



Task 3

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <string>

using namespace std;

struct student{

    string name;

    int rollNumber;

    int age;

};

void print14(student students[],int totalNum){

    cout<<"Names Of students with Age 14 : \n";

    for(int i=0;i<totalNum;i++){

        if(students[i].age==14){

            cout<<students[i].name<<"\n";

        }

    }

}

void printEven(student students[],int totalNum){

    cout<<"Names Of students with Even Roll Numbers : \n";

    for(int i=0;i<totalNum;i++){

        if(students[i].rollNumber%2==0){

            cout<<students[i].name<<"\n";

        }

    }

}

void searchDisplay(student students[],int totalNum){

    int inputNumber;

    cout<<"Roll number of student you want to search for : \n";

    cin>>inputNumber;

    for(int i=0;i<totalNum;i++){

        if(students[i].rollNumber==inputNumber){

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

            cout<<"Name : "<<students[i].name<<"\n";

            cout<<"Age : "<<students[i].age<<"\n";

            break;

        }

    }

}

int main(){

    srand(time(NULL));

    int totalStudents=11;

    student students[totalStudents];

    for(int i=0;i<totalStudents;i++){

        students[i].rollNumber=i+1;

        students[i].age=rand()%4+11;

        char nameForm[3]={char(97+i),char(122-i),'\0'};

        students[i].name=nameForm;

    }

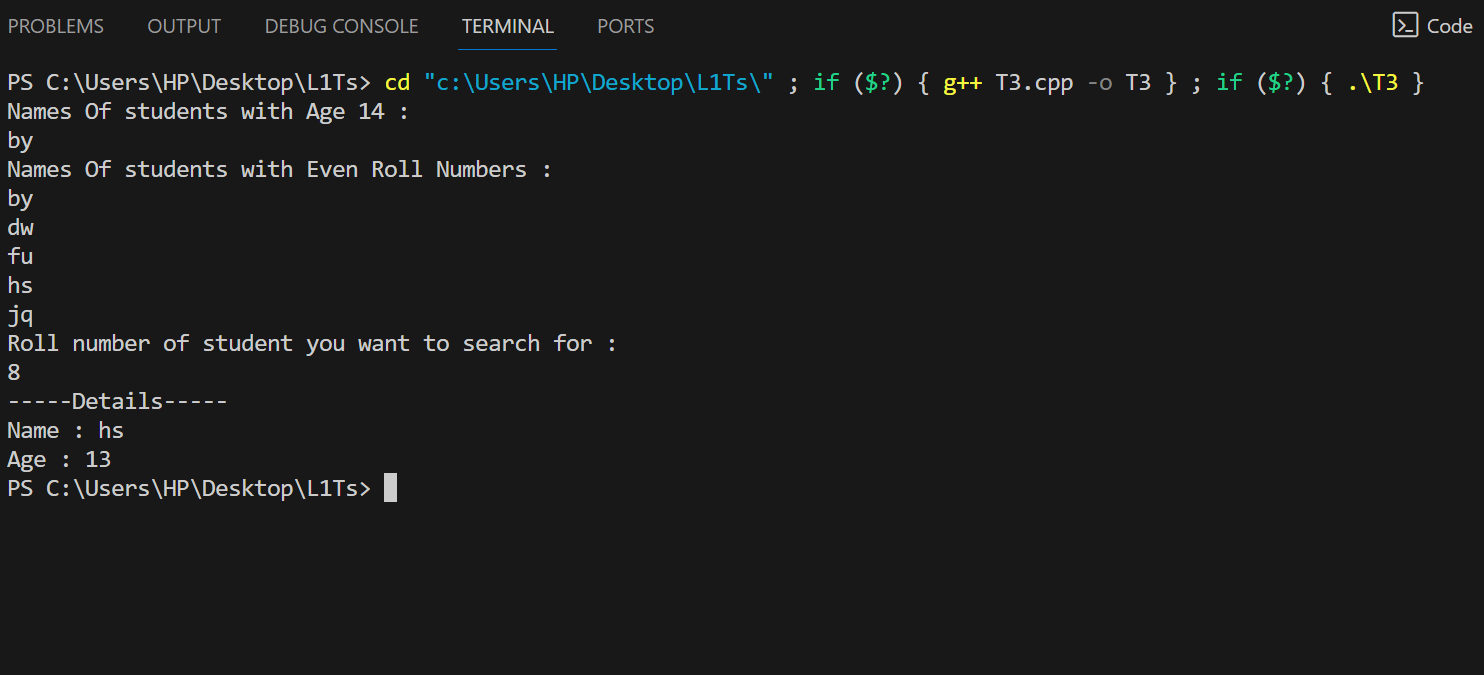
    print14(students,totalStudents);

    printEven(students,totalStudents);

    searchDisplay(students,totalStudents);

    return 0;

}



Task-4

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <string>

using namespace std;

struct employee{

    string name;

    float oldSalary;

    float newSalary;

    int workHours;

};

int main(){

    employee Employees[10];

    for(int i=0;i<10;i++){

        char nameForm[3]={char(97+i),char(122-i),'\0'};

        Employees[i].name=nameForm;

        Employees[i].oldSalary=rand()%900+100;

        Employees[i].workHours=(rand()%2==0?8:(rand()%2==0?10:rand()%13+12));

        Employees[i].newSalary=Employees[i].oldSalary+(Employees[i].workHours==8?50:Employees[i].workHours==10?100:150);

    }

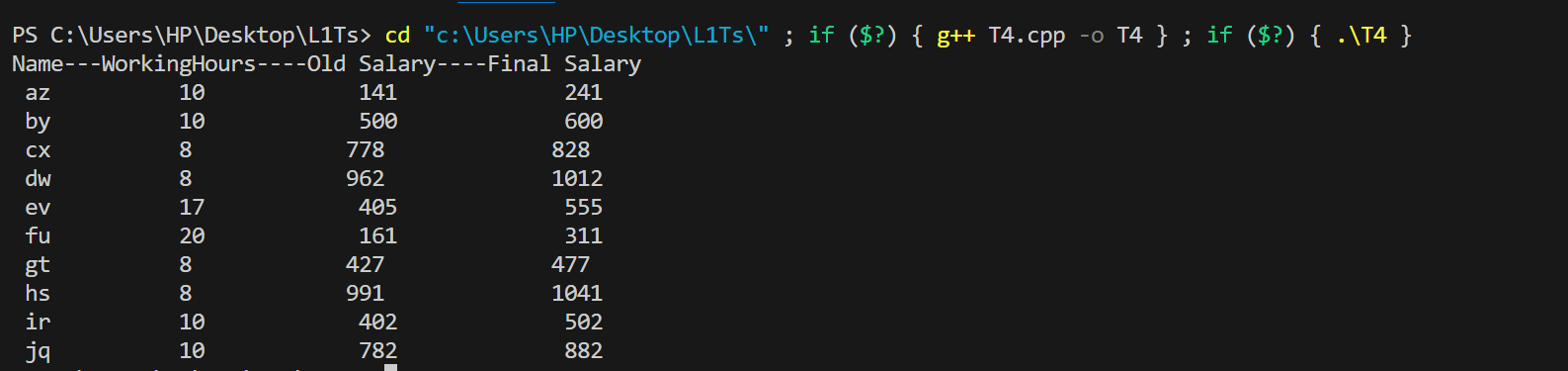
    cout<<"Name---WorkingHours----Old Salary----Final Salary\n";

    for(int i=0;i<10;i++){

        cout<<" "<<Employees[i].name<<"          "<<Employees[i].workHours<<"            "<<Employees[i].oldSalary<<"             "<<Employees[i].newSalary<<endl;

    }

}



Task-5

Code:

#include <iostream>

#include <conio.h>

using namespace std;

struct book{

    int accessNum;

    string auther;

    string title;

    int quantity;

    bool issued;

};

int menuDisplay(){

    int num;

    system("cls");

    cout<<"----Choose Your Option----\n";

    cout<<"1.Display Book Information\n";

    cout<<"2.Add New Book\n";

    cout<<"3.Display Books of a Specific Auther\n";

    cout<<"4.Display Books of a title\n";

    cout<<"5.Display Total Number of Books\n";

    cout<<"6.Issue a Book\n";

    cout<<"7.Exit Libruary\n";

    cout<<"Option : ";

    cin>>num;

    return num;

}

void displayInfo(book \*&books,int &totalBooks){

    system("cls");

    int aNum;

    cout<<"Enter Access Number of Book : ";

    cin>>aNum;

    for(int i=0;i<totalBooks;i++){

        if(books[i].accessNum==aNum){

            cout<<"\nBook Auther : "<<books[i].auther;

            cout<<"\nBook Title : "<<books[i].title;

            cout<<"\nBook Status : "<<(books[i].issued==true?"Issued":"Available");

            cout<<"\nBook Quantity : "<<books[i].quantity;

            break;

        }

    }

}

void addBook(book \*&books,int &totalBooks){

    book \*newBooks= new book[totalBooks+1];

    for(int i=0;i<totalBooks;i++){

        newBooks[i]=books[i];

    }

    newBooks[totalBooks].quantity=0;

    newBooks[totalBooks].issued=false;

    newBooks[totalBooks].accessNum=totalBooks+1;

    cout<<"\nEnter Book Auther : ";

    cin>>newBooks[totalBooks].auther;

    cout<<"\nEnter Book Title : ";

    cin>>newBooks[totalBooks].title;

    cout<<"\nThanks for the input";

    books=newBooks;

    totalBooks++;

    int max=0;

    for(int i=0;i<totalBooks;i++){

        if(books[i].title==books[totalBooks-1].title){

            max=(books[i].quantity>max?books[i].quantity:max);

        }

    }

    max++;

    for(int i=0;i<totalBooks;i++){

        if(books[i].title==books[totalBooks-1].title){

            books[i].quantity=max;

        }

    }

}

void displayBooksOfAuther(book \*&books,int &totalBooks){

    string aName;

    cout<<"Enter name of Auther : ";

    cin>>aName;

    for(int i=0;i<totalBooks;i++){

        if(aName==books[i].auther){

            cout<<"\nBook Title : "<<books[i].title;

            cout<<"\nBook Status : "<<(books[i].issued==true?"Issued":"Available");

            cout<<"\nAccess Number : "<<books[i].accessNum;

            cout<<"\nBook Quantity : "<<books[i].quantity;

            cout<<"\n\n";

        }

    }

}

void displayBooksOfTitle(book \*&books,int &totalBooks){

    string bTitle;

    cout<<"Enter Book title : ";

    cin>>bTitle;

    for(int i=0;i<totalBooks;i++){

        if(bTitle==books[i].title){

            cout<<"\nBook Auther : "<<books[i].auther;

            cout<<"\nBook Status : "<<(books[i].issued==true?"Issued":"Available");

            cout<<"\nAccess Number : "<<books[i].accessNum;

            cout<<"\nBook Quantity : "<<books[i].quantity;

            cout<<"\n\n";

        }

    }

}

void issueBook(book \*&books,int &totalBooks){

    int aNum;

    cout<<"Enter Access Number of Book : ";

    cin>>aNum;

    for(int i=0;i<totalBooks;i++){

        if(aNum==books[i].accessNum){

            cout<<"\nBook Auther : "<<books[i].auther;

            cout<<"\nAccess Title : "<<books[i].title;

            books[i].issued=true;

            cout<<"\nNew Book Status : "<<(books[i].issued==true?"Issued":"Available");

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

                if(books[i].title==books[j].title){

                    books[j].quantity--;

                }

            }

            cout<<"\nNew Book Quantity : "<<books[i].quantity;

        }

    }

}

int main(){

    book \*books;

    int totalBooks=0,choice;

    do{

        choice=menuDisplay();

        system("cls");

        switch(choice){

            case 1:

                displayInfo(books,totalBooks);

                break;

            case 2:

                addBook(books,totalBooks);

                break;

            case 3:

                displayBooksOfAuther(books,totalBooks);

                break;

            case 4:

                displayBooksOfTitle(books,totalBooks);

                break;

            case 5:

                cout<<"Total Number of Books : "<<totalBooks;

                break;

            case 6:

                issueBook(books,totalBooks);

                break;

            case 7:

                break;

            default :

                cout<<"Enter A Valid Choice";

        }

        cout<<"\npress anything to continue";

        getch();

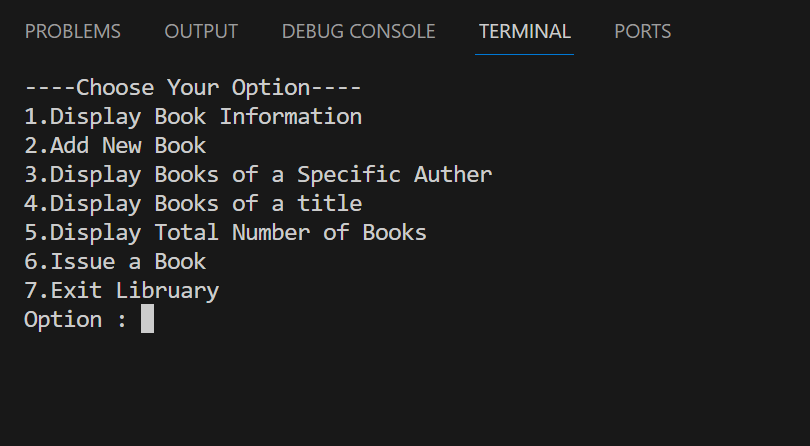
        system("cls");

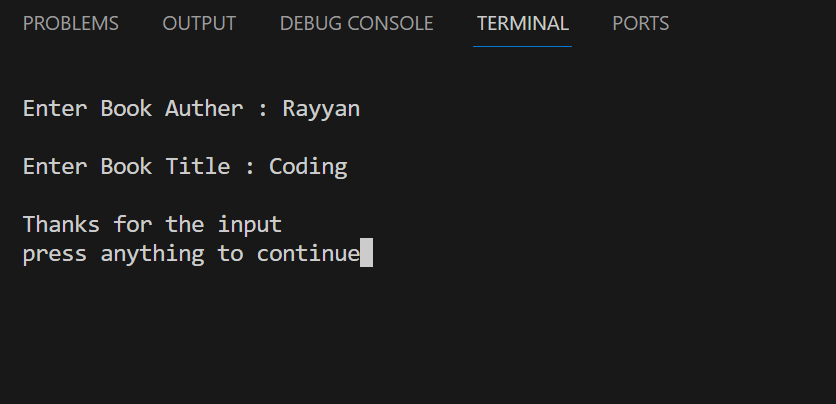
    }while(choice!=7);

    return 0;

}

Pictures :

 Menu

 Adding Book

A screen shot of a computer

Description automatically generatedAccessing Book

A screenshot of a computer

Description automatically generatedAdding The Same titleA screen shot of a computer

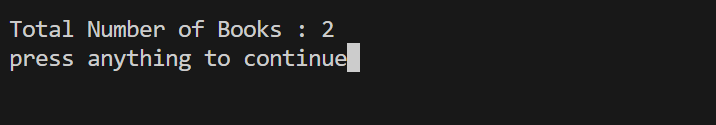
Description automatically generated Quantity increased for both

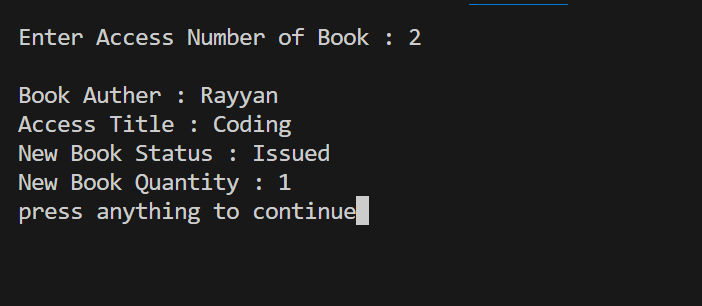
A screenshot of a computer program

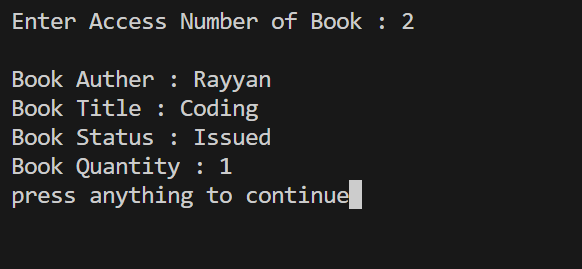
Description automatically generatedSearching by author

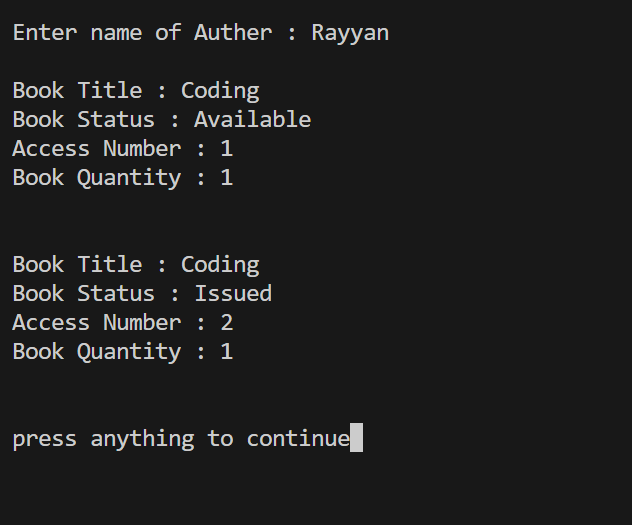
A screen shot of a computer

Description automatically generated Searching by book

 Total Number of Books

 Issuing Book 2

Displaying book info



Quantity for title decreased as one is issued now but one still available .Only access number 2 book is issued