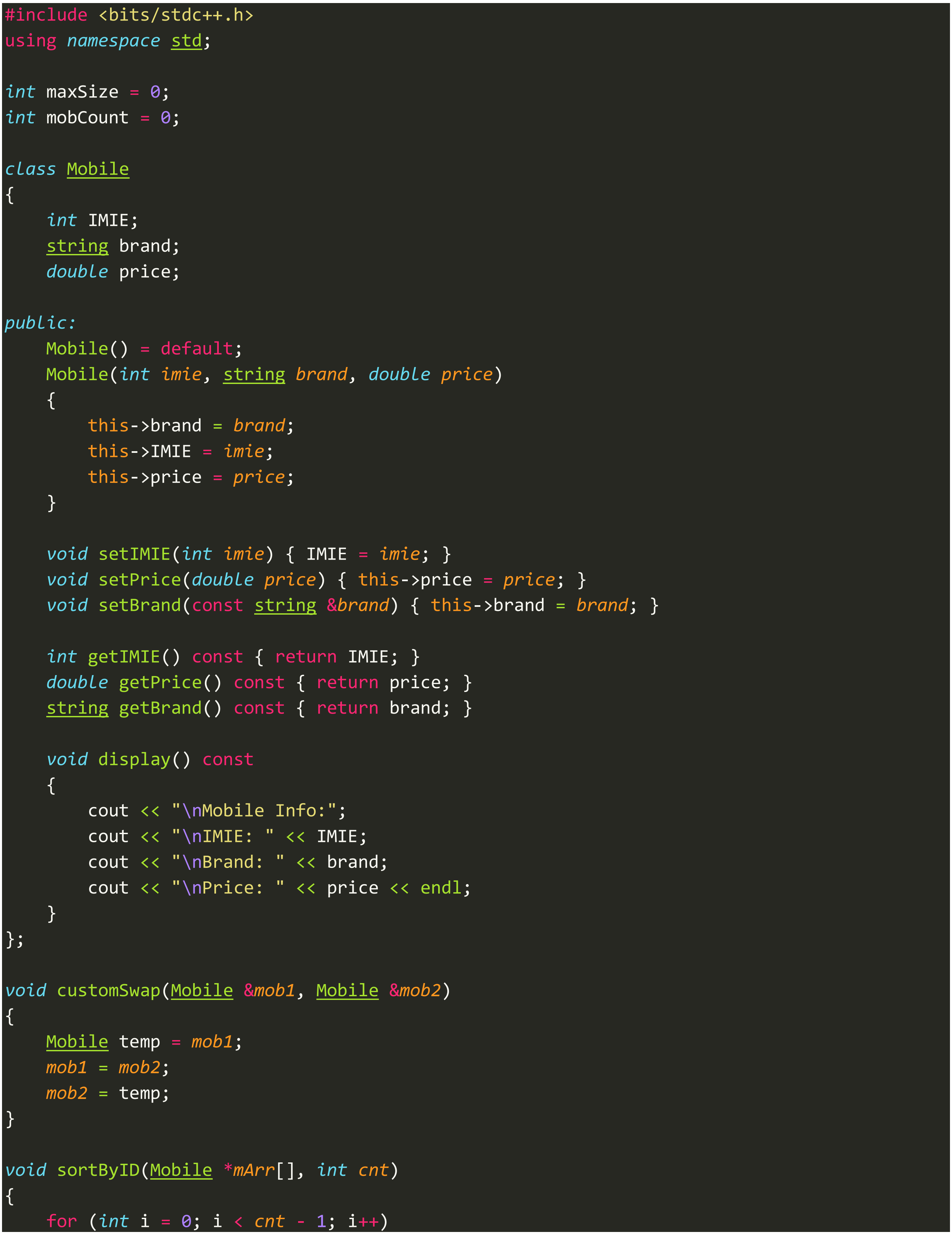
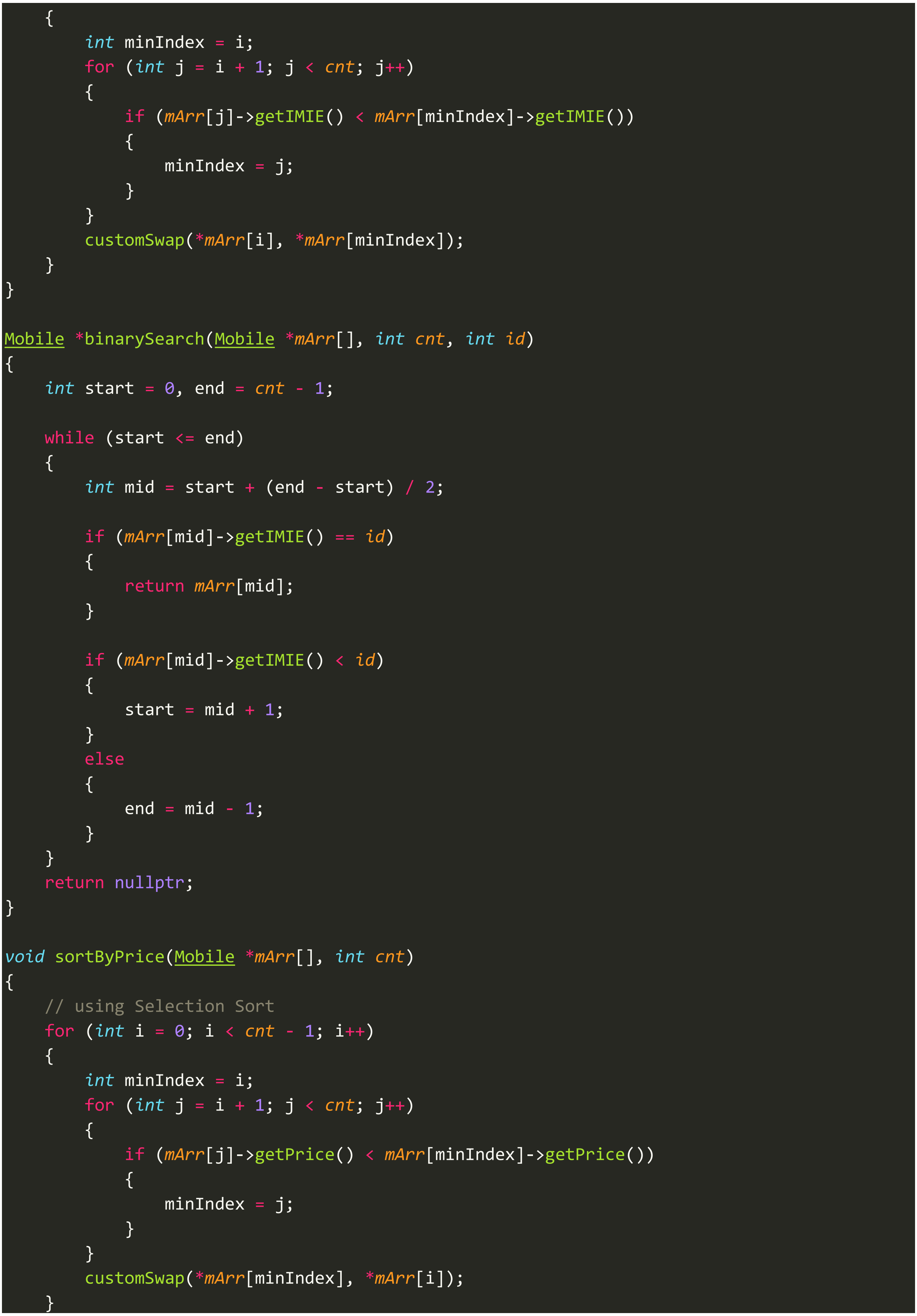
DSA Test (26/nov/24)

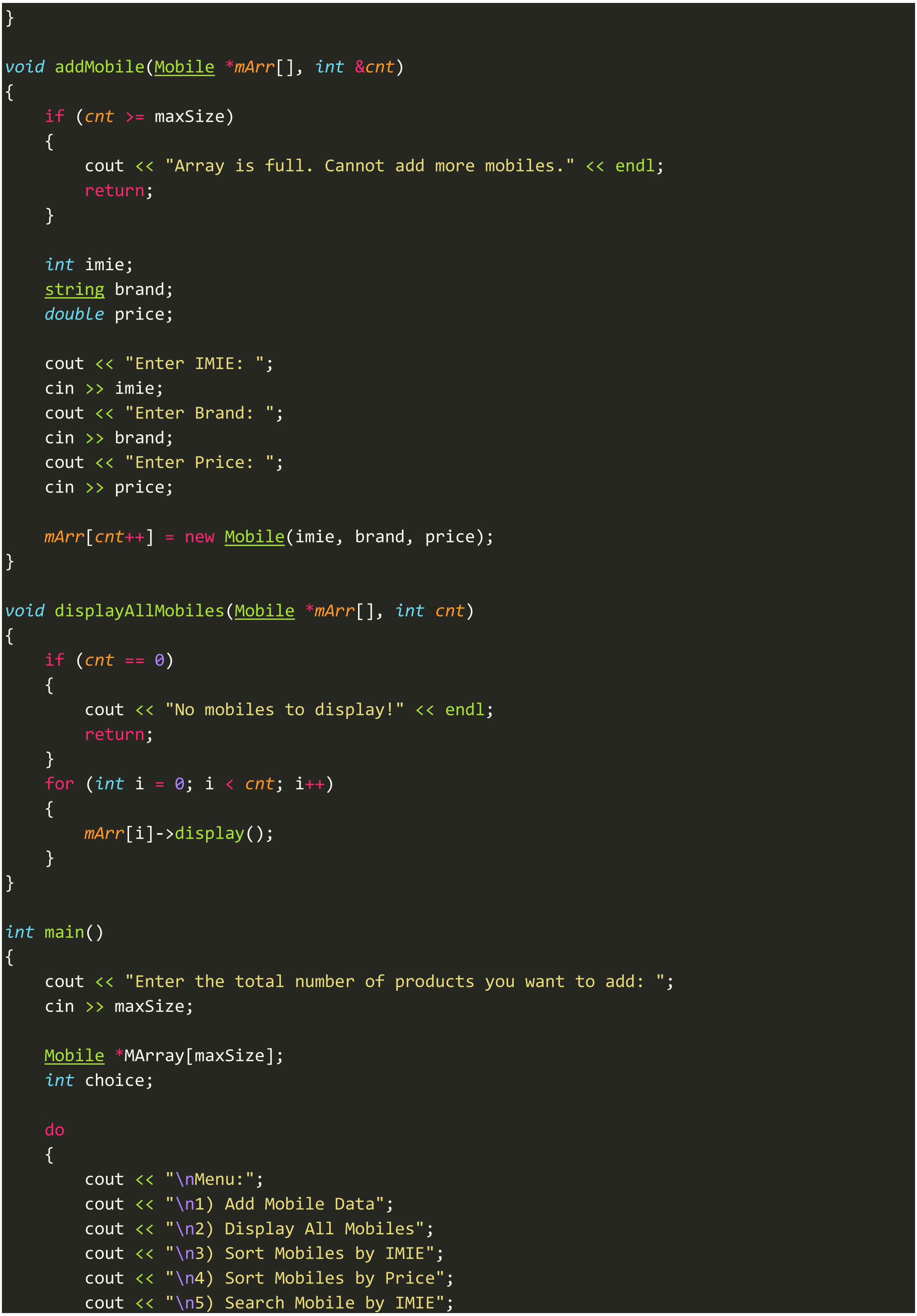
Q1:



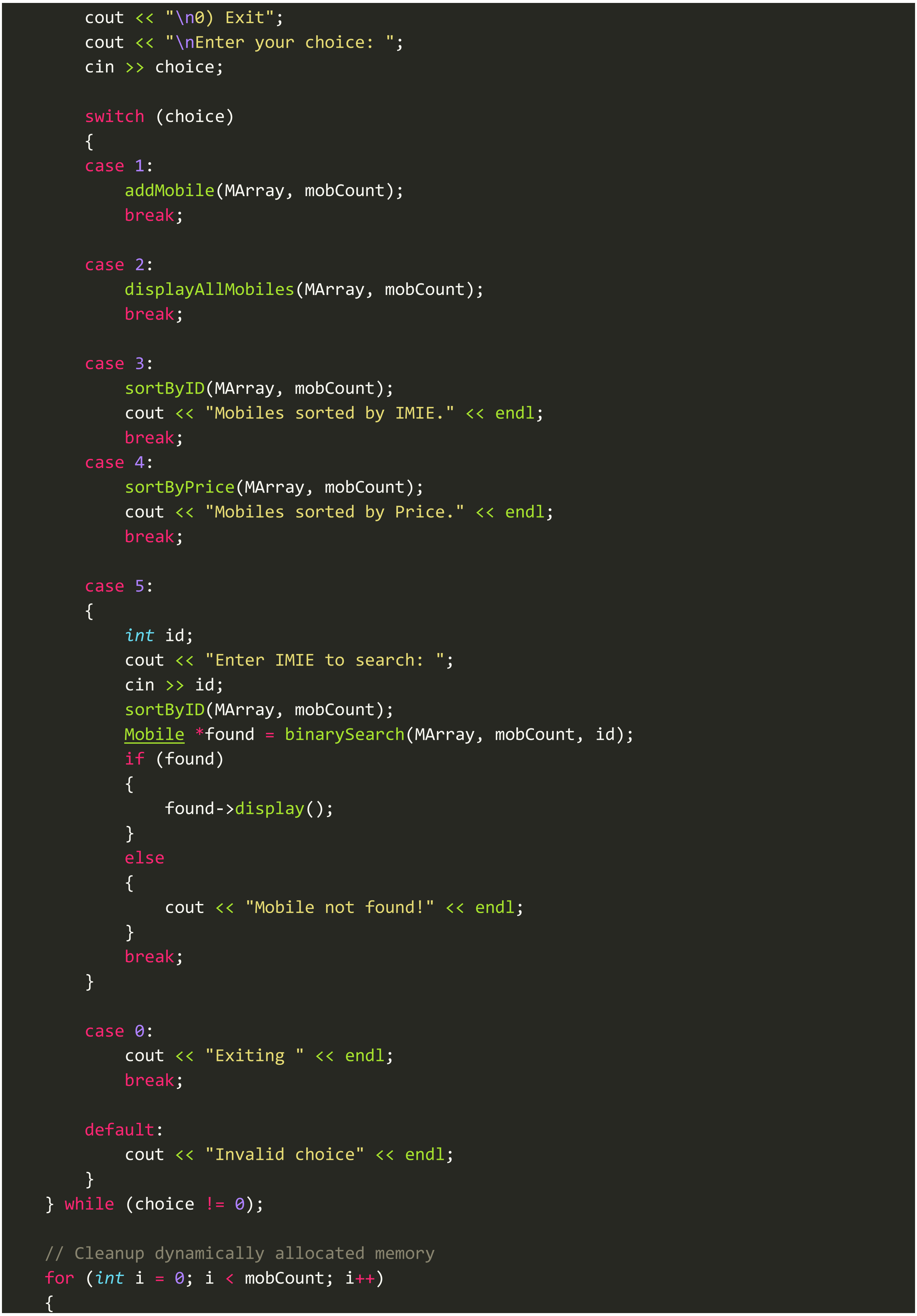
```cpp
#include <bits/stdc++.h>
using namespace std;

int maxSize = 0;
int mobCount = 0;

class Mobile
{
    int IMIE;
    string brand;
    double price;

public:
    Mobile() = default;
    Mobile(int imie, string brand, double price)
    {
        this->brand = brand;
        this->IMIE = imie;
        this->price = price;
    }

    void setIMIE(int imie) { IMIE = imie; }
    void setPrice(double price) { this->price = price; }
    void setBrand(const string &brand) { this->brand = brand; }

    int getIMIE() const { return IMIE; }
    double getPrice() const { return price; }
    string getBrand() const { return brand; }

    void display() const
    {
        cout << "\nMobile Info:";
        cout << "\nIMIE: " << IMIE;
        cout << "\nBrand: " << brand;
        cout << "\nPrice: " << price << endl;
    }
};

void customSwap(Mobile &mob1, Mobile &mob2)
{
    Mobile temp = mob1;
    mob1 = mob2;
    mob2 = temp;
}

void sortByID(Mobile *mArr[], int cnt)
{
    for (int i = 0; i < cnt - 1; i++)
```



```
    {
        int minIndex = i;
        for (int j = i + 1; j < cnt; j++)
        {
            if (mArr[j]->getIMIE() < mArr[minIndex]->getIMIE())
            {
                minIndex = j;
            }
        }
        customSwap(*mArr[i], *mArr[minIndex]);
    }
}

Mobile *binarySearch(Mobile *mArr[], int cnt, int id)
{
    int start = 0, end = cnt - 1;

    while (start <= end)
    {
        int mid = start + (end - start) / 2;

        if (mArr[mid]->getIMIE() == id)
        {
            return mArr[mid];
        }

        if (mArr[mid]->getIMIE() < id)
        {
            start = mid + 1;
        }
        else
        {
            end = mid - 1;
        }
    }
    return nullptr;
}

void sortByPrice(Mobile *mArr[], int cnt)
{
    // using Selection Sort
    for (int i = 0; i < cnt - 1; i++)
    {
        int minIndex = i;
        for (int j = i + 1; j < cnt; j++)
        {
            if (mArr[j]->getPrice() < mArr[minIndex]->getPrice())
            {
                minIndex = j;
            }
        }
        customSwap(*mArr[minIndex], *mArr[i]);
    }
```



```cpp
}

void addMobile(Mobile *mArr[], int &cnt)
{
    if (cnt >= maxSize)
    {
        cout << "Array is full. Cannot add more mobiles." << endl;
        return;
    }

    int imie;
    string brand;
    double price;

    cout << "Enter IMIE: ";
    cin >> imie;
    cout << "Enter Brand: ";
    cin >> brand;
    cout << "Enter Price: ";
    cin >> price;

    mArr[cnt++] = new Mobile(imie, brand, price);
}

void displayAllMobiles(Mobile *mArr[], int cnt)
{
    if (cnt == 0)
    {
        cout << "No mobiles to display!" << endl;
        return;
    }
    for (int i = 0; i < cnt; i++)
    {
        mArr[i]->display();
    }
}

int main()
{
    cout << "Enter the total number of products you want to add: ";
    cin >> maxSize;

    Mobile *MArray[maxSize];
    int choice;

    do
    {
        cout << "\nMenu:";
        cout << "\n1) Add Mobile Data";
        cout << "\n2) Display All Mobiles";
        cout << "\n3) Sort Mobiles by IMIE";
        cout << "\n4) Sort Mobiles by Price";
        cout << "\n5) Search Mobile by IMIE";
```



```
        cout << "\n0) Exit";
        cout << "\nEnter your choice: ";
        cin >> choice;

        switch (choice)
        {
        case 1:
            addMobile(MArray, mobCount);
            break;

        case 2:
            displayAllMobiles(MArray, mobCount);
            break;

        case 3:
            sortByID(MArray, mobCount);
            cout << "Mobiles sorted by IMIE." << endl;
            break;
        case 4:
            sortByPrice(MArray, mobCount);
            cout << "Mobiles sorted by Price." << endl;
            break;

        case 5:
        {
            int id;
            cout << "Enter IMIE to search: ";
            cin >> id;
            sortByID(MArray, mobCount);
            Mobile *found = binarySearch(MArray, mobCount, id);
            if (found)
            {
                found->display();
            }
            else
            {
                cout << "Mobile not found!" << endl;
            }
            break;
        }

        case 0:
            cout << "Exiting " << endl;
            break;

        default:
            cout << "Invalid choice" << endl;
        }
    } while (choice != 0);

    // Cleanup dynamically allocated memory
    for (int i = 0; i < mobCount; i++)
    {
        delete MArray[i];
    }

    return 0;
}
```

Output:

```
PS D:\Fullstack-Java-FirstBit-Solutions> & 'c:\Users\bhagv\.vscode\extensions\ms-
vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--
stdin=Microsoft-MIEngine-In-kkpjdicm.mdj' '--stdout=Microsoft-MIEngine-Out-
korud3lm.byn' '--stderr=Microsoft-MIEngine-Error-vdfdfppi.1bb' '--pid=Microsoft-
MIEngine-Pid-slnxt1vw.bmo' '--dbgExe=C:\TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'
Enter the total number of products you want to add: 5

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 1
Enter IMIE: 1234
Enter Brand: Saamsung
Enter Price: 12334

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 2

Mobile Info:
IMIE: 1234
Brand: Saamsung
Price: 12334

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 1
Enter IMIE:
124
Enter Brand: OnePluus
Enter Price:
4312

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 1
Enter IMIE: 7861
Enter Brand: Nokia
Enter Price: 200

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 11
Invalid choice

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 1
Enter IMIE: 100
Enter Brand: Motto
Enter Price: 19000

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 2

Mobile Info:
IMIE: 1234
Brand: Saamsung
Price: 12334

Mobile Info:
IMIE: 124
Brand: OnePluus
```

```
Price: 4312

Mobile Info:
IMIE: 7861
Brand: Nokia
Price: 200

Mobile Info:
IMIE: 100
Brand: Motto
Price: 19000

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 3
Mobiles sorted by IMIE.

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 2

Mobile Info:
IMIE: 100
Brand: Motto
Price: 19000

Mobile Info:
IMIE: 124
Brand: OnePluus
Price: 4312

Mobile Info:
IMIE: 1234
Brand: Saamsung
Price: 12334

Mobile Info:
IMIE: 7861
Brand: Nokia
Price: 200

Menu:
1) Add Mobile Data
2) Display All Mobiles
```

```
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice:
4
Mobiles sorted by Price.

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 2

Mobile Info:
IMIE: 7861
Brand: Nokia
Price: 200

Mobile Info:
IMIE: 124
Brand: OnePluus
Price: 4312

Mobile Info:
IMIE: 1234
Brand: Saamsung
Price: 12334

Mobile Info:
IMIE: 100
Brand: Motto
Price: 19000

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 5
Enter IMIE to search: 3423
Mobile not found!

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
```

```
0) Exit
Enter your choice: 5
Enter IMIE to search: 124

Mobile Info:
IMIE: 124
Brand: OnePluus
Price: 4312

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 5
Enter IMIE to search: 123
Mobile not found!

Menu:
1) Add Mobile Data
2) Display All Mobiles
3) Sort Mobiles by IMIE
4) Sort Mobiles by Price
5) Search Mobile by IMIE
0) Exit
Enter your choice: 0
Exiting
PS D:\Fullstack-Java-FirstBit-Solutions>
```

Q2:

Account.h

```cpp
#include <bits/stdc++.h>
using namespace std;

class Account
{
    static int countOfAcc;
    int accNo;
    double balance;

public:
    Account(int);

    bool withdrow(double);
    bool deposite(double);

    int getAccNo();
    int static getAccCount();
    double getBalance();
    void display();
};
```

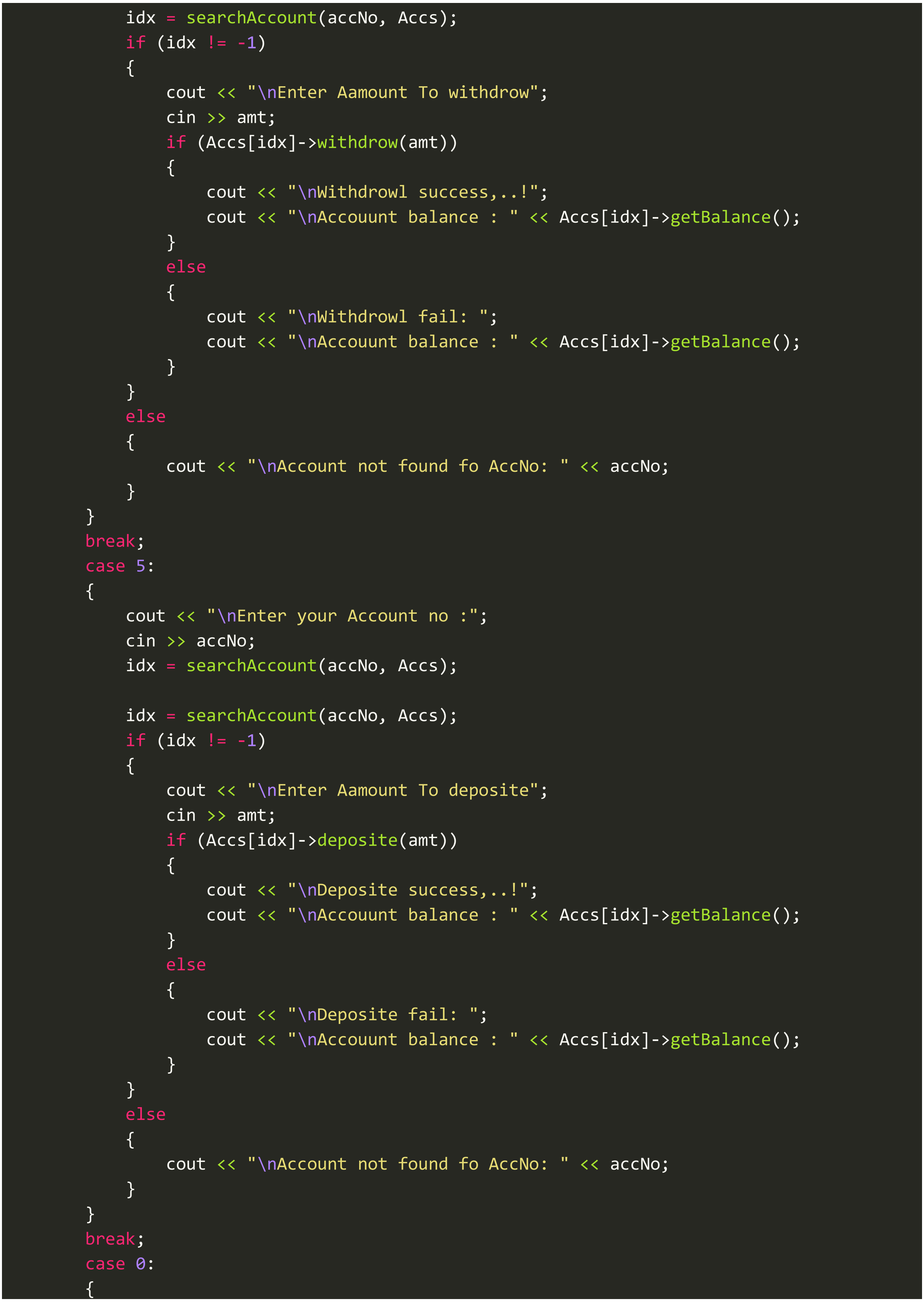
Account.cpp

```cpp
#include "Account.h"

int Account::countOfAcc = 0;

Account::Account(int AccNo)
{
    countOfAcc++;
    this->accNo = ((AccNo * AccNo) + Account::countOfAcc);
    this->balance = 0;
}

int Account::getAccNo()
{
    return this->accNo;
}
int Account::getAccCount()
{
    return countOfAcc;
}

double Account::getBalance()
{
    return this->balance;
}

bool Account::deposite(double amt)
```

```
{
    this->balance += amt;
    return true;
}

bool Account::withdrow(double amt)
{
    if (this->balance == 0.0 || this->balance < amt)
    {
        return false;
    }
    this->balance -= amt;
    return true;
}
void Account::display()
{
    cout << "\n========== Account Data ============";
    cout << "\nAccount No   :" << this->accNo;
    cout << "\nBalance      :" << this->balance;
    cout << "\nAccount ID   :" << countOfAcc;
    cout << "\n====================================\n";
}
```

Main.cpp

```
#include "Account.h"
int indx = -1;
int searchAccount(int AccNo, Account *ac[])
{
    for (int i = 0; i <= indx; i++)
    {
        if (ac[i]->getAccNo() == AccNo)
        {
            return i;
        }
    }

    return -1;
}
int main()
{
    int size, count = 0;
    cout << "\nEnter Size for arrray :";
    cin >> size;
    // Account acc(123);
    Account *Accs[size];

    int ch;
    int accNo, idx;
    double  amt;
    do
    {
        count = Account::getAccCount();
        cout << "\nMenu :\nNo of Accounts currently we have : " << count;
       cout << "\n1)Create Account : \t2) See Account details \n3) See Balance \t4) Withdrow
\n5) Deposite \t0) Exit ";
        cin >> ch;

        switch (ch)
        {
        case 1:
        {
            cout << "\nEnter Account no :";
            cin >> accNo;
            indx+=1;
            Accs[indx] = new Account(accNo);
            cout << "\nYour Account No is : " << Accs[indx]->getAccNo();
        }

        break;
        case 2:
        {
            cout << "\nEnter your Account no :";
            cin >> accNo;
            idx = searchAccount(accNo, Accs);
            if (idx != -1)
            {
                Accs[idx]->display();
            }
            else
            {
                cout << "\nAccount not found fo AccNo: " << accNo;
            }
        }
        break;
        case 3:
        {
            cout << "\nEnter your Account no :";
            cin >> accNo;
            idx = searchAccount(accNo, Accs);
            if (idx != -1)
            {
                cout << "\nAccouunt balance : " << Accs[idx]->getBalance();
            }
            else
            {
                cout << "\nAccount not found fo AccNo: " << accNo;
            }
        }
        break;
        case 4:
        {
            cout << "\nEnter your Account no :";
            cin >> accNo;
            idx = searchAccount(accNo, Accs);
```



```
            idx = searchAccount(accNo, Accs);
            if (idx != -1)
            {
                cout << "\nEnter Aamount To withdrow";
                cin >> amt;
                if (Accs[idx]->withdrow(amt))
                {
                    cout << "\nWithdrowl success,..!";
                    cout << "\nAccouunt balance : " << Accs[idx]->getBalance();
                }
                else
                {
                    cout << "\nWithdrowl fail: ";
                    cout << "\nAccouunt balance : " << Accs[idx]->getBalance();
                }
            }
            else
            {
                cout << "\nAccount not found fo AccNo: " << accNo;
            }
        }
        break;
        case 5:
        {
            cout << "\nEnter your Account no :";
            cin >> accNo;
            idx = searchAccount(accNo, Accs);

            idx = searchAccount(accNo, Accs);
            if (idx != -1)
            {
                cout << "\nEnter Aamount To deposite";
                cin >> amt;
                if (Accs[idx]->deposite(amt))
                {
                    cout << "\nDeposite success,..!";
                    cout << "\nAccouunt balance : " << Accs[idx]->getBalance();
                }
                else
                {
                    cout << "\nDeposite fail: ";
                    cout << "\nAccouunt balance : " << Accs[idx]->getBalance();
                }
            }
            else
            {
                cout << "\nAccount not found fo AccNo: " << accNo;
            }
        }
        break;
        case 0:
        {
```

```
                  cout  <<  "\n\n-----------------------------------------------------------
\nExiting!!!!!!!!!!";
        }
        break;
        default:
            cout << "\nInvalid Choice....!\n";
            break;
        }
    } while (ch != 0);

    return 0;
}
```

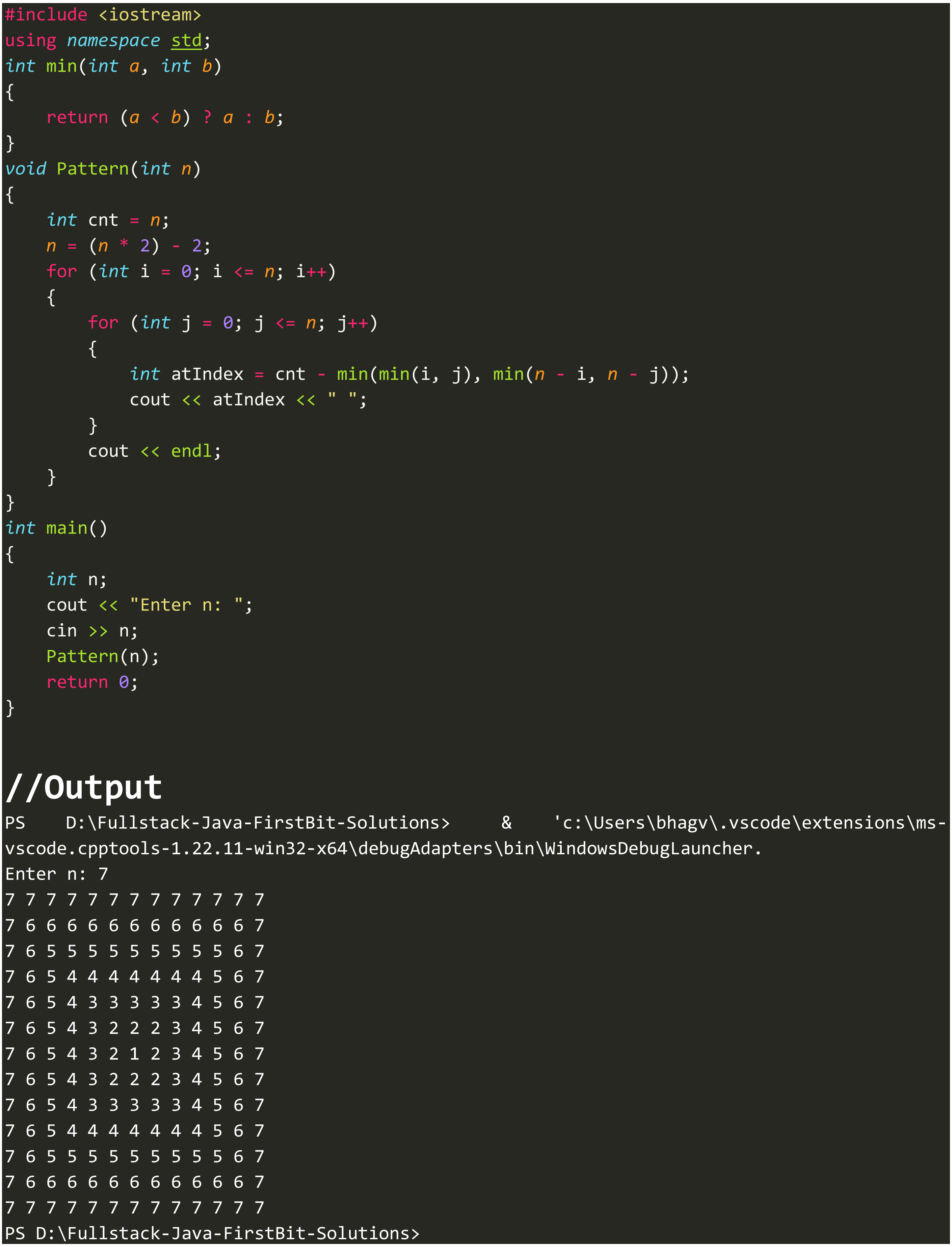
Output:

```
PS D:\Fullstack-Java-FirstBit-Solutions\DSA\Tests\26Nov\Account> g++ *.cpp -o main
PS D:\Fullstack-Java-FirstBit-Solutions\DSA\Tests\26Nov\Account> ./main
Enter Size for arrray :5
Menu :
No of Accounts currently we have : 0
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 1

Enter Account no :123
Your Account No is : 15130
Menu :
No of Accounts currently we have : 1
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 2

Enter your Account no :15130
========== Account Data ============
Account No   :15130
Balance      :0
Account ID   :1
====================================
Menu :No of Accounts currently we have : 1
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 5

Enter your Account no :15130
Enter Aamount To deposite12334.675

Deposite success,..!
Accouunt balance : 12334.7
Menu :
No of Accounts currently we have : 1
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 4
```

```
Enter your Account no :15130
Enter Aamount To withdrow234.34

Withdrowl success,..!
Accouunt balance : 12100.3
Menu :No of Accounts currently we have : 1
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 3

Enter your Account no :15130
Accouunt balance : 12100.3
Menu :
No of Accounts currently we have : 1
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 4

Enter your Account no :15130
Enter Aamount To withdrow2342143

Withdrowl fail:
Accouunt balance : 12100.3
Menu :
No of Accounts currently we have : 1
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 4

Enter your Account no :15130
Enter Aamount To withdrow12100

Withdrowl success,..!
Accouunt balance : 0.335
Menu :
No of Accounts currently we have : 1
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 2

Enter your Account no :12100

Account not found fo AccNo: 12100
Menu :
No of Accounts currently we have : 1
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 2

Enter your Account no :15130
========== Account Data ============
Account No   :15130
Balance      :0.335
Account ID   :1
```

```
====================================
Menu :
No of Accounts currently we have : 1
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 1

Enter Account no :23

Your Account No is : 531
Menu :
No of Accounts currently we have : 2
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 5

Enter your Account no :531
Enter Aamount To deposite34524523.345

Deposite success,..!
Accouunt balance : 3.45245e+07
Menu :No of Accounts currently we have : 2
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 2

Enter your Account no :531
========== Account Data ============
Account No   :531
Balance      :3.45245e+07
Account ID   :2
====================================
Menu :
No of Accounts currently we have : 2
1)Create Account :      2) See Account details
3) See Balance  4) Withdrow
5) Deposite     0) Exit 0
----------------------------------------------------------
Exiting!!!!!!!!!
PS D:\Fullstack-Java-FirstBit-Solutions\DSA\Tests\26Nov\Account>
```

Q3 :



```cpp
#include <iostream>
using namespace std;
int min(int a, int b)
{
    return (a < b) ? a : b;
}
void Pattern(int n)
{
    int cnt = n;
    n = (n * 2) - 2;
    for (int i = 0; i <= n; i++)
    {
        for (int j = 0; j <= n; j++)
        {
            int atIndex = cnt - min(min(i, j), min(n - i, n - j));
            cout << atIndex << " ";
        }
        cout << endl;
    }
}
int main()
{
    int n;
    cout << "Enter n: ";
    cin >> n;
    Pattern(n);
    return 0;
}
```

//Output

```
PS D:\Fullstack-Java-FirstBit-Solutions> & 'c:\Users\bhagv\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.
Enter n: 7
7 7 7 7 7 7 7 7 7 7 7 7 7
7 6 6 6 6 6 6 6 6 6 6 6 7
7 6 5 5 5 5 5 5 5 5 5 6 7
7 6 5 4 4 4 4 4 4 4 5 6 7
7 6 5 4 3 3 3 3 3 4 5 6 7
7 6 5 4 3 2 2 2 3 4 5 6 7
7 6 5 4 3 2 1 2 3 4 5 6 7
7 6 5 4 3 2 2 2 3 4 5 6 7
7 6 5 4 3 3 3 3 3 4 5 6 7
7 6 5 4 4 4 4 4 4 4 5 6 7
7 6 5 5 5 5 5 5 5 5 5 6 7
7 6 6 6 6 6 6 6 6 6 6 6 7
7 7 7 7 7 7 7 7 7 7 7 7 7
PS D:\Fullstack-Java-FirstBit-Solutions>
```