case 4:

bank.DisplayAllAccounts();

break;

case 5:

Console.WriteLine("Exiting the system...");

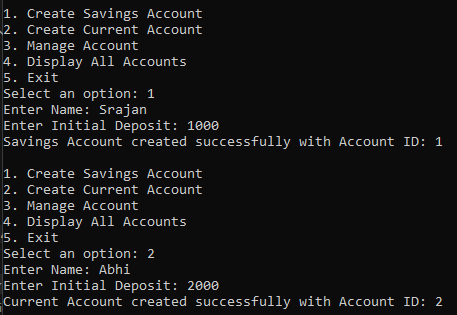
Environment.Exit(1);

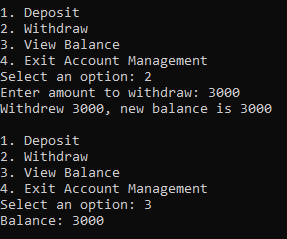
break;

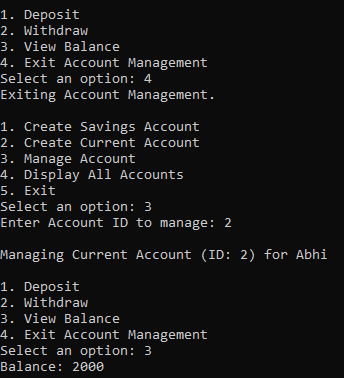
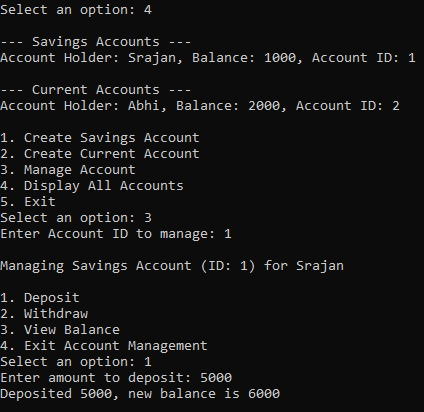
default:

Console.WriteLine("Invalid option. Try again.");

break;

 } } } }



****

**Q. Bus Reservation System**

class BusReservation {

public int[][] Seats;

public static void Main(String[] args) {

BusReservation obj = new BusReservation();

obj.InitializeSeating();

while (true) {

Console.WriteLine("Choose : ");

Console.WriteLine("1)Book Seat ");

Console.WriteLine("2)Cancel Seat Reservation");

Console.WriteLine("3)View Seating Arrangement");

Console.WriteLine("4)Exit");

int choice = int.Parse(Console.ReadLine()!);

switch (choice) {

case 1:

obj.BookSeat(); break;

case 2:

obj.cancelSeat(); break;

case 3:

obj.showSeats(); break;

case 4:

Environment.Exit(1);break;

default:

Console.WriteLine("Wrong Choice!!"); break;

} } }

public void InitializeSeating() {

Console.WriteLine("Enter number of rows:");

int numRows = int.Parse(Console.ReadLine()!);

Seats = new int[numRows][];

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

Console.WriteLine($"Enter number of seats for row {i + 1}:");

int numSeats = int.Parse(Console.ReadLine()!);

Seats[i] = new int[numSeats];

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

Seats[i][j] = 0; // not booked

} } }

public void BookSeat() {

showSeats();

Console.WriteLine("Enter the row no : ");

int row = int.Parse(Console.ReadLine()!);

Console.WriteLine("Enter the Column no. : ");

int col = int.Parse(Console.ReadLine()!);

if (row >= 0 && row < Seats.Length &&

col >= 0 && col < Seats[row].Length)

{

Seats[row-1][col-1] = 1;

showSeats();

}

**Q. WAP to enter a string, convert it to uppercase and add 2 in each character present in the string and display original and modified string.**

class Program {

public static void Main(string[] args) {

Console.WriteLine("Enter a string : ");

string x = Console.ReadLine()!;

Console.WriteLine($"The original String : {x}");

x = x.ToUpper();

string ans = "";

for (int i = 0; i < x.Length; i++) {

if (x[i] == ' ') {

ans += ' '; // keep spaces consistent

continue;

}

if (x[i] + 2 > 'Z') {

ans += (char)((x[i] + 2) - 26);

}

else {

ans += (char)(x[i] + 2);

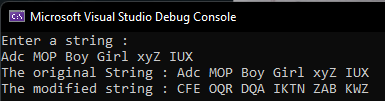
}

}

Console.WriteLine($"The modified string : {ans}");

}

}



**Q. Input a name, convert each character to its opposite case and print the original and modified names along with suitable messages.**

class Program {

public static void Main(string[] args) {

Console.WriteLine("Enter name : ");

string x = Console.ReadLine()!;

string result = "";

for (int i = 0; i < x.Length; i++) {

if (char.IsUpper(x[i])) {

result += char.ToLower(x[i]);

}

else if (char.IsLower(x[i])) {

result += char.ToUpper(x[i]);

}

else {

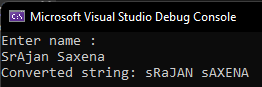
result += x[i]; // keeping non-alpha chars unchanged

} }

Console.WriteLine("Converted string: " + result);

}

}



**Q. WAP to accept a word and convert to lowercase and display new word by replacing only the vowels with the character that follows it.**

class Program {

public static void Main(string[] args) {

Console.WriteLine("Enter a string : ");

string x = Console.ReadLine()!;

x = x.ToLower();

string ans = "";

for (int i = 0; i < x.Length; i++) {

if (x[i] == 'a' || x[i] == 'e' || x[i] == 'i' || x[i] == 'o' || x[i] == 'u')

{

ans += (char)(x[i] + 1);

}

else {

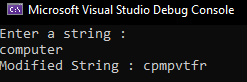
ans += x[i];

} }

Console.WriteLine($"Modified String : {ans}");

}

}



**Q. WAP to create a class ArrangeLetters and method Arrage() with a string parameter. Convert the word in uppercase and arrange each letter of the word in alphabetical order. Print the word before and after arranging the letters in A-Z order.**

class ArrangeLetters {

public static void Main(string[] args) {

Arrange("bca");

}

public static void Arrange(string x) {

Console.WriteLine($"Given : {x}");

x = x.ToUpper();

Console.WriteLine($"In Upper Case : {x}");

char[] arr = x.ToCharArray();

for(int i = 0; i < arr.Length - 1; i++) {

for(int j = 0; j < arr.Length - i - 1; j++) {

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

char temp = arr[j]; // swap

arr[j] = arr[j + 1];

arr[j + 1] = temp;

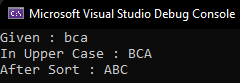
} } }

String ans = new String(arr);

Console.WriteLine($"After Sort : {ans}");

}

}



**Q. WAP to create an interface Encryptable with methods encrypt (String data) and decrypt (String encryptedData) that define encryption and decryption operations. Create a class AES that implements the Encryptable interface and provide their own encryption and decryption algorithms.**

public interface Encryptable {

string Encrypt(string data);

string Decrypt(string encryptedData);

}

public class AES : Encryptable {

private int shift;

public AES(int shiftValue) {

shift = shiftValue;

}

public string Encrypt(string data) {

char[] encrypted = new char[data.Length];

for (int i = 0; i < data.Length; i++) {

char c = data[i];

if (char.IsLetter(c)) {

char d = char.IsUpper(c) ? 'A' : 'a';

encrypted[i] = (char)((c + shift - d) % 26 + d);

}

else {

encrypted[i] = c;

} }

return new string(encrypted);

}

public string Decrypt(string encryptedData) {

char[] decrypted = new char[encryptedData.Length];

for (int i = 0; i < encryptedData.Length; i++) {

char c = encryptedData[i];

if (char.IsLetter(c)){

char d = char.IsUpper(c) ? 'A' : 'a';

decrypted[i] = (char)((c - shift - d + 26) % 26 + d);

}

else {

decrypted[i] = c;

} }

return new string(decrypted);

}}

class Program {

static void Main(string[] args) {

while (true) {

Console.WriteLine("\n--- Encryption Program ---");

Console.WriteLine("1. Encrypt using AES (Shift Cipher)");

Console.WriteLine("2. Decrypt using AES (Shift Cipher)");

Console.WriteLine("3. Exit");

Console.Write("Choose an option: ");

int choice = int.Parse(Console.ReadLine()!);

if (choice == 1) {

Console.Write("Enter text to encrypt: ");

string text = Console.ReadLine()!;

Console.Write("Enter shift value: ");

int shift = int.Parse(Console.ReadLine()!);

AES aes = new AES(shift);

string encryptedText = aes.Encrypt(text);

Console.WriteLine("Encrypted Text: " + encryptedText);

}

else if (choice == 2)

{

Console.Write("Enter text to decrypt: ");

string text = Console.ReadLine()!;

Console.Write("Enter shift value: ");

int shift = int.Parse(Console.ReadLine()!);

AES aes = new AES(shift);

string decryptedText = aes.Decrypt(text);

Console.WriteLine("Decrypted Text: " + decryptedText);

}

else if (choice == 3) {

Console.WriteLine("Exiting program...");

break;

}

else {

Console.WriteLine("Invalid choice, try again.");

} } } }

