Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Омский государственный технический университет»

Кафедра «Информатика и вычислительная техника»

Отчёт по лабораторной работе № 1

по дисциплине

«Проектирование и тестирование программного обеспечения»

|  |  |
| --- | --- |
|  | Выполнил  Студент гр. МО-181  Грачев Д. В.    (подп., дата)  Проверил  Старший преподаватель каф. ИВТ  Звонов А.О.    (подп., дата) |

Омск 2020

**Задача 1.3 (LCD-дисплей)**

РС/UVa IDs: 110104/706

Ваш друг только недавно купил себе новый компьютер. До этого самой мощной машиной, которую он когда-либо использовал, был карманный калькулятор. Он немного расстроен, потому что LCD-дисплей его калькулятора ему правился больше, чем экран его компьютера. Чтобы обрадовать его, напишите программу, которая отображает числа в стиле LCD-дисплея.

Входные данные

Входные данные содержат несколько строк, по строке для каждого числа, которое нужно отобразить. Каждая строка содержит целые числа s и n, где n - это число, которое нужно отобразить (0 ≤ n ≤ 99 999 999) и s, это размер, в котором оно должно быть изображено (0 ≤ s ≤ 10). Входные данные заканчиваются строкой, состоящей из двух нулей, которую не нужно обрабатывать.

Выходные данные

Выведите числа, указанные во входных данных в стиле LCD-дисплея, используя s «-» символов для горизонтальных сегментов и s «|» символов для вертикальных. Каждая цифра занимает ровно s + 2 колонок и 2 s + 3 строк. Не забудьте заполнить все белое пространство, занимаемое цифрами, пробелами, включая последнюю цифру. Между двумя цифрами должна быть точно одна колонка пробелов. Выводите пустую строку после каждого числа.

Ниже представлена программа:

namespace L11

{

class Program

{

static Digit zero = new Digit() { name = 0, top = "-", b1l = "|", b1r = "|", mid = " ", b2l = "|", b2r = "|", bot = "-" };

static Digit one = new Digit() { name = 1, top = " ", b1l = " ", b1r = "|", mid = " ", b2l = " ", b2r = "|", bot = " " };

static Digit two = new Digit() { name = 2, top = "-", b1l = " ", b1r = "|", mid = "-", b2l = "|", b2r = " ", bot = "-" };

static Digit three = new Digit() { name = 3, top = "-", b1r = "|", mid = "-", b2r = "|", bot = "-" };

static Digit four = new Digit() { name = 4, b1l = "|", b1r = "|", mid = "-", b2r = "|"};

static Digit five = new Digit() { name = 5, top = "-", b1l = "|", mid = "-", b2r = "|", bot = "-" };

static Digit six = new Digit() { name = 6, top = "-", b1l = "|", mid = "-", b2l = "|", b2r = "|", bot = "-" };

static Digit seven = new Digit() { name = 7, top = "-", b1r = "|", b2r = "|" };

static Digit eight = new Digit() { name = 8, top = "-", b1l = "|", b1r = "|", mid = "-", b2l = "|", b2r = "|", bot = "-" };

static Digit nine = new Digit() { name = 9, top = "-", b1l = "|", b1r = "|", mid = "-", b2r = "|", bot = "-" };

static Digit[] digits = new Digit[] { zero, one, two, three, four, five, six, seven, eight, nine };

static void Main()

{

try

{

string[] input = File.ReadAllLines("file.txt"); //Считывание всех строк файла

foreach (string line in input)

{

try

{

int s = Convert.ToInt32(line.Split(new char[] { ' ' })[0]); // Размер цифр

int n = Convert.ToInt32(line.Split(new char[] { ' ' })[1]); // Число для визуализации

int[] digs = GetDigitsArray(n); // Получение массива, содержащего цифры числа

List<Digit> ds = new List<Digit>();

foreach (int a in digs)

foreach (Digit dig in digits)

if (dig.name == a) ds.Add(dig); // Добавление в коллекцию цифр для визуализации тех цифр, что содержатся в файле

Draw(ds.ToArray(), s); // Визуализировать цифры

Console.WriteLine();

}

catch (FormatException Fe)

{

Console.WriteLine(Fe.Message);

}

}

}

catch (IOException IOe)

{

Console.WriteLine(IOe.Message);

}

}

static int[] GetDigitsArray(int n)

{

List<int> digs = new List<int>();

int d = 10;

while(n > 0)

{

digs.Add(n % d);

n /= 10;

}

digs.Reverse();

return digs.ToArray();

}

static void Draw(Digit[] digs, int s)

{

DrawTop(digs, s);

DrawB1(digs, s);

DrawMid(digs, s);

DrawB2(digs, s);

DrawBot(digs, s);

Console.WriteLine();

}

static void DrawTop(Digit[] digs, int s)

{

foreach(var dig in digs)

{

Console.Write(" ");

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

{

Console.Write(dig.top);

}

Console.Write(" ");

Console.Write(" ");

}

}

static void DrawB1(Digit[] digs, int s)

{

Console.WriteLine();

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

{

foreach (var dig in digs)

{

Console.Write(dig.b1l);

for (int j = 0; j < s; j++)

Console.Write(" ");

Console.Write(dig.b1r);

Console.Write(" ");

}

Console.WriteLine();

}

}

static void DrawMid(Digit[] digs, int s)

{

foreach (var dig in digs)

{

Console.Write(" ");

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

{

Console.Write(dig.mid);

}

Console.Write(" ");

Console.Write(" ");

}

}

static void DrawB2(Digit[] digs, int s)

{

Console.WriteLine();

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

{

foreach (var dig in digs)

{

Console.Write(dig.b2l);

for (int j = 0; j < s; j++)

Console.Write(" ");

Console.Write(dig.b2r);

Console.Write(" ");

}

Console.WriteLine();

}

}

static void DrawBot(Digit[] digs, int s)

{

foreach (var dig in digs)

{

Console.Write(" ");

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

{

Console.Write(dig.bot);

}

Console.Write(" ");

Console.Write(" ");

}

}

class Digit

{

public int name;

public string top = " ";

public string b1l = " ";

public string b1r = " ";

public string mid = " ";

public string b2l = " ";

public string b2r = " ";

public string bot = " ";

}

}

}

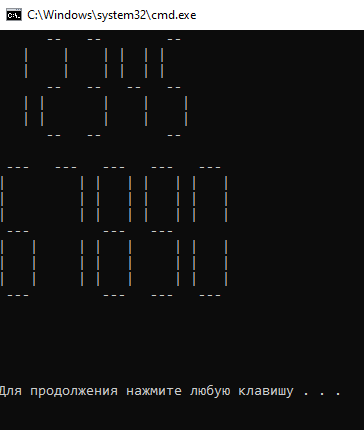
Выбранный путь решения задачи:

1. Для каждой строки файла, считывание данных

2. Создание коллекции цифр, которые нужно визуализировать   
3. Визуализация цифр

Используемые структуры данных для решения задачи: списки, массивы, классы

Результат выполнения программы для входных данных 2 12345, 3 67890, 0 0:



**Задача 2.3 (По порядку)**

PC/UVaIDs: 110205/10205

В Большом городе мною казино. В одном из них дилер жульничает. Она довела до совершенства несколько перетасовочных трюков; каждый трюк меняет порядок карт одним и тем же образом, когда бы он ни был использован. Простой пример - это трюк «нижней карты», при котором нижняя карта переносится на верх колоды. Используя различные комбинации известных ей трюков, нечестный дилер может расположить карты практически в любом желаемом порядке.

Вы были наняты менеджером службы безопасности, чтобы поймать этого дилера. Вам дали список всех трюков, используемых дилером, а также список визуальных подсказок, благодаря которым вы можете определить, какой трюк она использует в любой данный момент времени. Ваша задача состоит в том, чтобы предсказать порядок карт после последовательности трюков.

Стандартная игральная колода состоит из 52 карт, четыре масти по 13 значений каждая. Значения карт: 2, 3. 4, 5, б, 7, 8, 9, 10, Валет (Jack), Дама (Queen), Король (King), Туз (Асе). Названия мастей: Трефы (Clubs), Бубны (Diamonds), Червы (Hearts), Пики (Spades). Определенная карта в колоде может быть однозначно определена по ее значению и масти и обычно обозначается <значение> of <масть>. Например, «9 of Hearts» или «King of Spades». По традиции карты в новой колоде идут по мастям в алфавитном порядке, а потом по значению в порядке, описанном выше.

Входные данные

Входные данные начинаются со строки, содержащей одно положительное целое число, которое означает количество тестовых блоков, за которой следует пустая строка. Между двумя последовательными тестовыми блоками также находится пустая строка.

Каждый блок состоит из целого числа n < 100 - числа трюков, которые знает дилер. Далее следуют n наборов по 52 целых числа, каждый из которых содержит все числа от 1 до 52 в определенном порядке. Внутри каждого набора из 52 чисел, если i стоит j-м по счету, то это значит, что трюк перемещает i-ю карту в колоде на j-е место (не важно какая именно это была карта).

Далее следуют несколько строк, каждая из которых содержит целое число k от 1 до n. Оно означает, что вы увидели, как дилер делает k-й трюк, заданный во входных данных.

Выходные данные

Для каждого тестового блока считайте, что дилер начинает с новой колоды, карты в которой расположены в исходном порядке, описанном выше. Выведите новый порядок карт в колоде, после того как все трюки были проделаны. Выходные данные для двух последовательных блоков должны быть разделены пустой строкой.

Ниже представлена программа:

Файл Program.cs

namespace L12

{

class Program

{

static void Main()

{

try

{

using (StreamReader stream = new StreamReader("file.txt"))

{

int blocks = Convert.ToInt32(stream.ReadLine());

stream.ReadLine();

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

{

int n = Convert.ToInt32(stream.ReadLine());

int[][] tricks = new int[n][];

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

tricks[j] = stream.ReadLine().Split(new char[] { ' ' }).Select(int.Parse).ToArray();

string input;

List<int> usedTricks = new List<int>();

while ((input = stream.ReadLine()) != null)

usedTricks.Add(Convert.ToInt32(input));

Card[] set = Card.GenerateSet();

for (int j = 0; j < tricks.Length; j++)

{

if (usedTricks.Contains(j + 1))

set = Trick(set, tricks[j]);

}

string result = string.Join("\n", set.Select(x => x.Info()).ToArray()) + "\n";

Console.Write(result);

Console.WriteLine("---------------");

}

}

}

catch(IOException IOe)

{

Console.WriteLine(IOe.Message);

}

catch(FormatException Fe)

{

Console.WriteLine(Fe.Message);

}

}

static Card[] Trick(Card[] set, int[] trick)

{

Card[] newSet = new Card[set.Length];

for (int i = 0; i < 52; i++)

{

var trickCardNumb = trick[i] - 1;

newSet[i] = set[trickCardNumb];

}

return newSet;

}

}

}

Файл Card.cs

namespace L12

{

class Card

{

private CardTypes Type { get; set; }

private Suits Suit { get; set; }

public Card(int number)

{

number--;

Type = (CardTypes)(number % 13);

Suit = (Suits)(number / 13);

}

public string Info () => Type.ToString().Remove(0, 1) + " of " + Suit.ToString();

public static Card[] GenerateSet()

{

Card[] cards = new Card[52];

for (int i = 1; i <= 52; i++)

cards[i - 1] = new Card(i);

return cards;

}

}

}

Файл CardTypes.cs

namespace L12.Enums

{

enum CardTypes

{

t2, t3, t4, t5, t6, t7, t8, t9,

t10, tJack, tQueen, tKing, tAce

}

}

Файл Suits.cs

namespace L12.Enums

{

enum Suits

{

Clubs, Diamonds,

Hearts, Spades

}

}

Выбранный путь решения задачи:

1. Считываем данные тестового блока  
2. Генерируем набор карт в стандартном порядке  
3. Для каждого замеченного трюка, вызываем функцию Trick, которая меняет порядок карт в соответствии с трюком

4. Выводим результат

Используемые структуры данных для решения задачи: списки, массивы, классы, перечисления.

Результат работы программы при следующих входных данных:

1

2

2 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 52 51

52 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 1

1

2



Выводы:

В процессе выполнения лабораторной работы пришло понимание того, что предварительное продумывание проектирования программы сильно помогает в процессе ее написания, также как и прописывание всех переменным и функциям их значение в программе.