КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра програмних систем і технологій

Захист програм та даних

Практична робота № 2.1

Виконали студенти

групи ІПЗ-33

Щука Богдан

Ємець Євгенія

Олійник Анастасія

Перевірила

доцент Супрун О.М.

Київ-2018

**Хід роботи**

**Завдання 1**

У практичній роботі необхідно по алгоритму MD5 отримати хеш-образ повідомлення, що складається з перших трьох букв своєї сім'ї.

Лістинг програми

**using** System;

**using** System.Linq;  
**using** System.Text;  
**using** System.Numerics;  
  
**namespace** MD5{  
 **internal class** Program{  
 **private static** BigInteger Func1(BigInteger x, BigInteger y, BigInteger z) {  
 **var** value = (x & y) | (~x & z);  
 **return** value; }  
  
 **private static** BigInteger Func2(BigInteger x, BigInteger y, BigInteger z) {  
 **var** value = (x & z) | (~z & y);  
 **return** value; }  
  
 **private static** BigInteger Func3(BigInteger x, BigInteger y, BigInteger z) {  
 **var** value = x ^ y ^ z;  
 **return** value; }  
  
 **static** BigInteger Func4(BigInteger x, BigInteger y, BigInteger z) {  
 **var** value = y ^ (~z | x);  
 **return** value; }  
  
 **private static** BigInteger RotateLeft(BigInteger n, **int** bits) {  
 **return** (n << bits) | (n >> (32 - bits)); }  
  
 **private static void** Md5 (**string** message) {  
 **var** messageInAscii = Encoding.**ASCII**.GetBytes(message);  
 **var** newMessage = messageInAscii.Aggregate(**""**, (current, t) => current + Convert.ToString(t, 2));  
  
 **var** messageLengthInBytes = newMessage.**Length**;  
  
 newMessage += **"1"**;  
 **while** (newMessage.**Length** % 512 != 448) {  
 newMessage += **"0"**; }  
 **var** messageLengthInBytes64 = Convert.ToString(messageLengthInBytes, 2).PadLeft(64, **'0'**);  
 newMessage += messageLengthInBytes64.Substring(32, 32);  
 newMessage += messageLengthInBytes64.Substring(0, 32);  
  
 **uint** A = 0x01234567;  
 **uint** B = 0x89ABCDEF;  
 **uint** C = 0xFEDCBA98;  
 **uint** D = 0x76543210;  
  
 BigInteger AA = A;  
 BigInteger BB = B;  
 BigInteger CC = C;  
 BigInteger DD = D;  
  
 BigInteger Divisor = 4294967296;  
  
 BigInteger[] T =  
 {  
 0xd76aa478, 0xe8c7b756, 0x242070db, 0xc1bdceee,  
 0xf57c0faf, 0x4787c62a, 0xa8304613, 0xfd469501,  
 0x698098d8, 0x8b44f7af, 0xffff5bb1, 0x895cd7be,  
 0x6b901122, 0xfd987193, 0xa679438e, 0x49b40821,  
 0xf61e2562, 0xc040b340, 0x265e5a51, 0xe9b6c7aa,  
 0xd62f105d, 0x02441453, 0xd8a1e681, 0xe7d3fbc8,  
 0x21e1cde6, 0xc33707d6, 0xf4d50d87, 0x455a14ed,  
 0xa9e3e905, 0xfcefa3f8, 0x676f02d9, 0x8d2a4c8a,  
 0xfffa3942, 0x8771f681, 0x6d9d6122, 0xfde5380c,  
 0xa4beea44, 0x4bdecfa9, 0xf6bb4b60, 0xbebfbc70,  
 0x289b7ec6, 0xeaa127fa, 0xd4ef3085, 0x04881d05,  
 0xd9d4d039, 0xe6db99e5, 0x1fa27cf8, 0xc4ac5665,  
 0xf4292244, 0x432aff97, 0xab9423a7, 0xfc93a039,  
 0x655b59c3, 0x8f0ccc92, 0xffeff47d, 0x85845dd1,  
 0x6fa87e4f, 0xfe2ce6e0, 0xa3014314, 0x4e0811a1,  
 0xf7537e82, 0xbd3af235, 0x2ad7d2bb, 0xeb86d391  
 };  
  
 **string** block;  
 **var** X = **new string**[16];  
 **int** index;  
 **int**[] K = {  
 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,  
 1, 6, 11, 0, 5, 10, 15, 4, 9, 14, 3, 8, 13, 2, 7, 12,  
 5, 8, 11, 14, 1, 4, 7, 10, 13, 0, 3, 6, 9, 12, 15, 2,  
 0, 7, 14, 5, 12, 3, 10, 1, 8, 15, 6, 13, 4, 11, 2, 9  
 };  
 **var** counter = 0;  
   
 **for** (**var** i = 0; i < newMessage.**Length** / 512; i++) {  
 block = newMessage.Substring(0, 512);  
 index = 0;  
 **for** (**int** j = 0; j < X.**Length**; j++) {  
 X[j] = newMessage.Substring(index, 32);  
 index += 32; }  
  
 **for** (**var** j = 0; j < 4; j++) {  
 **switch** (j) {  
 **case** 0:  
 Console.WriteLine(**"1st round"**);  
 **break**;  
 **case** 1:  
 Console.WriteLine(**"2nd round"**);  
 **break**;  
 **case** 2:  
 Console.WriteLine(**"3rd round"**);  
 **break**;  
 **case** 3:  
 Console.WriteLine(**"4th round"**);  
 **break**;  
 }  
 **for** (**var** k = 0; k < 4; k++) {  
 **switch** (j) {  
 **case** 0:  
 AA = BigInteger.Remainder(BB + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(AA + Func1(BB, CC, DD), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 7)), Divisor);  
 counter++;   
 Console.WriteLine(**"AA = "** + **string**.Format(**"0x{0:x2} "**, AA));  
  
 DD = BigInteger.Remainder(AA + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(DD + Func1(AA, BB, CC), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 12)), Divisor);  
 counter++;  
 Console.WriteLine(**"DD = "** + **string**.Format(**"0x{0:x2} "**, DD));  
  
 CC = BigInteger.Remainder(DD + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(CC + Func1(DD, AA, BB), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 17)), Divisor);  
 counter++;  
 Console.WriteLine(**"CC = "** + **string**.Format(**"0x{0:x2} "**, CC));  
  
 BB = BigInteger.Remainder(CC + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(BB + Func1(CC, DD, AA), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 22)), Divisor);  
 counter++;  
 Console.WriteLine(**"BB = "** + **string**.Format(**"0x{0:x2}\n"**, BB));  
 **break**;  
  
 **case** 1:  
 AA = BigInteger.Remainder(BB + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(AA + Func1(BB, CC, DD), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 5)), Divisor);  
 counter++;  
 Console.WriteLine(**"AA = "** + **string**.Format(**"0x{0:x2} "**, AA));  
 DD = BigInteger.Remainder(AA + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(DD + Func1(AA, BB, CC), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 9)), Divisor);  
 counter++;  
 Console.WriteLine(**"DD = "** + **string**.Format(**"0x{0:x2} "**, DD));  
  
 CC = BigInteger.Remainder(DD + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(CC + Func1(DD, AA, BB), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 14)), Divisor);  
 counter++;  
 Console.WriteLine(**"CC = "** + **string**.Format(**"0x{0:x2} "**, CC));  
  
 BB = BigInteger.Remainder(CC + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(BB + Func1(CC, DD, AA), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 20)), Divisor);  
 counter++;  
 Console.WriteLine(**"BB = "** + **string**.Format(**"0x{0:x2}\n"**, BB));  
 **break**;  
  
 **case** 2:  
 AA = BigInteger.Remainder(BB + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(AA + Func1(BB, CC, DD), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 4)), Divisor);  
 counter++;  
 Console.WriteLine(**"AA = "** + **string**.Format(**"0x{0:x2} "**, AA));  
  
 DD = BigInteger.Remainder(AA + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(DD + Func1(AA, BB, CC), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 11)), Divisor);  
 counter++;  
 Console.WriteLine(**"DD = "** + **string**.Format(**"0x{0:x2} "**, DD));  
  
 CC = BigInteger.Remainder(DD + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(CC + Func1(DD, AA, BB), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 16)), Divisor);  
 counter++;  
 Console.WriteLine(**"CC = "** + **string**.Format(**"0x{0:x2} "**, CC));  
  
 BB = BigInteger.Remainder(CC + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(BB + Func1(CC, DD, AA), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 23)), Divisor);  
 counter++;  
 Console.WriteLine(**"BB = "** + **string**.Format(**"0x{0:x2}\n"**, BB));  
 **break**;  
  
 **case** 3:  
 AA = BigInteger.Remainder(BB + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(AA + Func1(BB, CC, DD), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 6)), Divisor);  
 counter++;  
 Console.WriteLine(**"AA = "** + **string**.Format(**"0x{0:x2} "**, AA));  
  
 DD = BigInteger.Remainder(AA + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(DD + Func1(AA, BB, CC), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 10)), Divisor);  
 counter++;  
 Console.WriteLine(**"DD = "** + **string**.Format(**"0x{0:x2} "**, DD));  
  
 CC = BigInteger.Remainder(DD + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(CC + Func1(DD, AA, BB), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 15)), Divisor);  
 counter++;  
 Console.WriteLine(**"CC = "** + **string**.Format(**"0x{0:x2} "**, CC));  
  
 BB = BigInteger.Remainder(CC + (RotateLeft((BigInteger.Remainder(BigInteger.Remainder(BigInteger.Remainder(BB + Func1(CC, DD, AA), Divisor) + BigInteger.Parse(X[K[counter]]), Divisor) + T[counter], Divisor)), 21)), Divisor);  
 counter++;  
 Console.WriteLine(**"BB = "** + **string**.Format(**"0x{0:x2}\n"**, BB));  
 **break**;  
 }   
 }   
 }  
  
 A = (**uint**)(BigInteger.Remainder(A + AA, Divisor));  
 B = (**uint**)(BigInteger.Remainder(B + BB, Divisor));  
 C = (**uint**)(BigInteger.Remainder(C + CC, Divisor));  
 D = (**uint**)(BigInteger.Remainder(D + DD, Divisor));  
  
 newMessage = newMessage.Substring(512);  
 }  
  
 Console.Write(**"MD5 = "** +   
 **string**.Format(**"{0:x2}"**, A & 0xffff) +  
 **string**.Format(**"{0:x2}"**, A >> 16) +  
 **string**.Format(**"{0:x2}"**, B & 0xffff) +  
 **string**.Format(**"{0:x2}"**, B >> 16) +  
 **string**.Format(**"{0:x2}"**, C & 0xffff) +  
 **string**.Format(**"{0:x2}"**, C >> 16) +  
 **string**.Format(**"{0:x2}"**, D & 0xffff) +  
 **string**.Format(**"{0:x2}"**, D >> 16));  
 }  
  
 **static void** Main(**string**[] args)  
 {  
 **var** message = **"luz"**;  
 Md5(message);  
 Console.ReadKey();  
 }  
 }  
}

Приклад роботи програми

