1 Задачи.

1. Дана колода из 36 игральных карт (это наш алфавит). Каково количество информации, измеренной в битах, заключено в сообщении:

а) вытащенная из колоды карта является дамой пик;

log2(36/1)=5.17 bit

б) вытащенная из колоды карта является бубновой;

log2(4/1)=2 bit

в) вытащенная из колоды карта старше десятки;

log2(36/10)=1.17 bit

г) вытащенная из колоды карта чёрной масти;

log2(2/2)=1 bit

д) вытащенная из колоды карта является тузом.

log2(36/4)=3,16 bit

е) вытащенная из колоды карта является тузом или дамой красной масти;

log2(36/6)=2,58 bit

ё) из колоды вытащили пиковый туз, а потом из оставшейся колоды вытащили валет червей;

1/36\*1/35\*1/1260, log2(1260/1)=10,29 bit

ж) из колоды вытащили туз, а потом из оставшейся колоды вытащили семёрку;

4/36\*4/35=4/315, log2(315/4)=6,29 bit

з) из колоды вытащили шестёрку красной масти, потом из оставшейся колоды вытащили семёрку, а потом из оставшейся колоды вытащили шестёрку пик.

2/36\*435\*1/34=1/5355

log2(5355/1)=12,38 bit

2. Дан мешок, в котором лежат 3 чёрных шара, 5 красных и 7 белых шаров (это наш алфавит). Каково количество информации, измеренной в натах, заключено в сообщении:

а) вытащенный из мешка шар является чёрным;

ln (15/3)=1,6 nat

б) вытащенный из мешка шар является чёрным или белым;

ln(15/10)= 0,4 nat

в) из мешка вытащили красный шар, а потом ещё один красный шар;

5/15\*4/14 ln(10,5) = 2,35

г) из мешка вытащили красный шар, а потом не красный шар;

5/15\*10/14 = 50/210, ln(4,2)=1,43

д) из мешка вытащили красный шар, а потом не белый шар.

5/15\*7/14 = 35/210, ln(6)=1,79

3. Алфавит источника сообщений {а,б,в}. Источник сообщений посылает нам слова.

а) сколько трит информации содержится в слове «аааа», полученном из источника сообщений?

4

б) сколько трит информации содержится в слове «бааб», полученном из источника сообщений?

4

в) сколько трит информации содержится в слове из четырёх символов, полученных из источника сообщений?

4

г) сколько бит информации содержится в слове «вбаа», полученном из источника сообщений?

8

д) в каком случае а) или г) мы получили большее количество информации?

г

4. Какой длины должно быть сообщение в алфавите {а,б,в,г}, содержащее 16 бит информации?

4

5. Сколько символов должно быть в алфавите, из которого составлено слово, длиной четыре символа, содержащее 32 бита информации?

256

2 Задачи.

1. Запишите десятичное число 5 в системе счисления с основанием 1.

|||||5

1. Какое наибольшее двоичное вещественное число, содержащее 4 разряда для дробной части, не превосходит значение 1,0?

0,1111

1. Каково наименьшее положительное шестнадцатеричное вещественное число, содержащее 4 разряда для дробной части?

0,0001

1. Какие максимальные целые числа представимы в 32-разрядном, 64-разрядном и 128-разрядном компьютерах?

4 294 967 295

18 446 744 073 709 551 616

3 402 82 366 920 938 463 463 374 607 431 768 211 455

6. Переведите десятичные вещественные числа в двоичные:

а) 8,125 = 1000.001

б) 7,625 = 111.101

в) 10,875 = 1010.111

г) 15,5078125 = 1111.1000001

3 Задачи.

1. Найдите однобайтовый двоичный код для чисел. Разрешается использовать калькулятор Windows.

а) 11 = 10112

б) 128 =100000002

в) 19= 100112

2. Найдите десятичное число по его двоичному коду. Разрешается использовать калькулятор Windows.

а) 11001011= 203

б) 10000001=129

в) 11111111=255

г) 1000000000000000=32768

д) 1000000000000001=32769

е) 1111111111111111=65535

3. В каких случаях сложение чисел x и y переполнит разрядную сетку и в переменной z будет получен неправильный результат:

|  |  |
| --- | --- |
| а) | byte x = 0b10000000;  byte y = 0b01111111;  byte z = (byte)(x + y);  z=255 |
| б) | byte x = 0b10000001;  byte y = 0b01111111;  byte z = (byte)(x + y);  z=0 неправильный результат |
| в) | byte x = 0b10000001;  byte y = 0b01111111;  uint z = (uint)(x + y);  z=256 |
| г) | byte x = 0x0F;  byte y = 0xF0;  byte z = (byte)(x + y);  z=255 |
| д) | byte x = 0x99;  byte y = 0x99;  byte z = (byte)(x + y);  z=50 |
| е) | byte x = 0x7F;  byte y = 0x81;  byte z = (byte)(x + y);  z = 0 неправильный результат |
| ж) | ushort x = 0xFFFE;  ushort y = 0b1;  ushort z = (ushort)(x + y);  z=65535 |
| з) | ushort x = 0xF0F0;  ushort y = 0x0F0F;  ushort z = (ushort)(x + y);  z=65535 |
| и) | ushort x = 0xFFFF;  ushort z = (ushort)(x++);  z=65535 неправильный результат |
| к) | ushort x = 0xFFFF;  uint z = (uint)(x++);  z=65535 |

4. Объясните смысл преобразования данных в двух вариантах программы

А- преобразует возвращенное значение функцией в тип byte

Б- функция выбросит исключение т.к. х+1 происходит переполнения при выполнении арифметических операций

|  |  |
| --- | --- |
| а) | byte x = byte.MaxValue;  byte z = (byte) checked(x + 1);  Console.WriteLine(z); |
| б) | byte x = byte.MaxValue;  byte z = checked((byte) (x + 1));  Console.WriteLine(z); |

8. Написать программу для вывода на экран в двоичном и десятичном виде максимального и минимального чисел типа uint и ulong.

Console.WriteLine("минимальное значение ulong = {0}, максимальное значение ulong = {1}", ulong.MinValue, ulong.MaxValue);

Console.WriteLine("минимальное значение uint = {0}, максимальное значение uint = {1}", uint.MinValue, uint.MaxValue);

4 Задачи.

1. Разработать программу побитного выведения беззнакового байта на экран. Использовать умножение на маску и сдвиги.

static void f4\_1(byte x)

{

var list = new Boolean[8];

for (int i = 0; i < 8; i++)

{

list[i] = ((x & (1 << i)) != 0);

}

Array.Reverse(list);

for (int i = 0; i < 8; i++)

{

Console.Write(list[i] == false ? 0 : 1);

}

}

1. Разработать программу определения того, что биты, расположенные на двух средних позициях введённого с клавиатуры байта, одинаковы.

static void f4\_2(byte x)

{

Console.Write((x & (1 << 3)) == (x & (1 << 4)) ? "Биты одинаковы" : "Биты не одинаковы");

}

4. Написать программу проверки равенства старшей и младшей тетрад 2-байтовой переменной, задаваемой в программе в двоичном виде.

static void f4\_4(int x)

{

for (int i = 0, k = 7; i < 4; i++, k--)

{

if (Convert.ToBoolean(x & (1 << i)) != Convert.ToBoolean(x & (1 << k)))

{

Console.Write("старшая и младшая тетрада не равны");

return;

}

}

Console.Write("старшая и младшая тетрада равны");

}

5 Написать программу проверки того, что 2-байтовая переменная содержит одинаковые байты.

Console.WriteLine("Введите 8 битное двоичное число");

string str = Console.ReadLine();

for (int i = 0, k = 7; i < 4; i++, k--)

{

if (str[i] != str[k])

{

Console.WriteLine(" Не является полиндромом");

return;

}

}

Console.WriteLine(" Является полиндромом");

5 Задачи.

1. Найдите дополнительный код указанного числа. Разрешается использовать калькулятор Windows в режиме «Программист» и устанавливать в нём разрядность числа.

а) -11 → однобайтовое знаковое число; дополнительный код 0101

б) -128 → двухбайтовое знаковое число; дополнительный код 1000 0000

в) -19 → четырёхбайтовое знаковое число; дополнительный код 01101

г) -129 → двухбайтовое знаковое число; дополнительный код 0111 111

д) -1 → четырёхбайтовое знаковое число.

2. Найдите десятичное число по его дополнительному коду. Разрешается использовать калькулятор Windows в режиме «Программист». Не забывайте устанавливать в нём правильную разрядность числа и правильно вводить код числа.

а) 1100 = 12

б) 1111 = 15

в) 110011 = 51

г) 11001011 = 53

д) 10000001 = 127

е) 11111111 = 1

ж) 1011111101 = 765

з) 1000000000000000 = -32768

и) 1000000000000001= -32767

к) 1111111111111111 = -1

л) 110010111100101111001011 = 13355979

м) 11001011110010111100101111001100 = -875836468

3. Напишите программу перевода двоичного числа к заданным типам и выведите ответ в десятичной форме:

byte a = 0b11001011;

sbyte b = 0b11001011;

Console.WriteLine("byte ={0:D}, sbyte={1:D}",a,b);

а) 11001011 → byte и sbyte

б) 11111111 → short и ushort

в) 1100101110000001 → byte и sbyte

г) 1111111100000000 → byte и sbyte

д) 1111111100000000 → short и ushort

4. В программе задано число. Реализовать в программе вывод его на экран в двоичном коде заданной длины. Объяснить полученные результаты.

а) sbyte x = -128; // вывести 8-битный код

б) sbyte x = -1; // вывести 64-битный код

в) short x = -129; // вывести 16-битный код

г) int x = 0x7fffffff; // вывести 16-битный код

д) int x = 0x7fffffff; // вывести 32-битный код

е) int x = 0x7fff0000; // вывести 8-битный код

//sbyte x = -128,y = 8;

//sbyte x = -1, y = 64;

//short x = -129, y = 16;

int x = 0x7fffffff, y = 16;

//int x = 0x7fffffff, y = 32;

//int x = 0x7fff0000, y = 8;

var list = new Boolean[y];

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

{

list[i] = ((x & (1 << i)) != 0);

}

Array.Reverse(list);

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

{

Console.Write(list[i] == false ? 0 : 1);

}

Console.WriteLine();

1. Написать программу для вывода на экран в двоичном и десятичном виде максимального и минимального чисел типа long.

Console.WriteLine("минимальное значение long = {0}, максимальное значение long = {1}", long.MinValue, long.MaxValue);

6 Задачи.

1. Найти сумму вручную столбиком:

а)

11010101

01101010

11101010

00111010

1001100011

б)

11111001

11111010

11101011

01111011

1101011001

в)

00111010

01001010

10001011

10111111

111001110

2. Найти значение флагов CF и OF при ручном выполнении операций над 4-разрядными беззнаковыми числами. Результат операции полагать знаковым 4-разрядным числом и записать его в двоичном и десятичном виде.

а) 0101-1101 cf 0 of 1

б) 0101+1011 cf 1 of 0

в) 0111-0011 cf 0 of 0

г) 1111+1000 cf 1 of 0

д) 1111-0010 cf 0 of 0

е) 0100+0110 cf 0 of 0

г) 0000-0010 cf 0 of 1

3. Найти значение флагов CF и OF при ручном выполнении операций над 4-разрядными знаковыми числами. Результат операции полагать знаковым 4-разрядным числом и записать его в двоичном и десятичном виде.

а) 0101-1101 = 0010 cf 1 of 0

б) 0101+1011 = 1010 cf 0 of 0

в) 0111-0011= 0100 cf 0 of 0

г) 1111+1000 = 1111 cf 0 of 0

д) 1111-0010 = 1101 cf 1 of 0

е) 0100+0110 = 0010 cf 1 of 0

г) 0000-0010 = 1010 cf 0 of 0

5. Подобрать такие 16-разрядные знаковые значения для чисел A и B, при которых выполняются условия:

а) A-B приводит к OF=0 A=2 B=1

б) A+B приводит к OF=0 A=2 B=1

в) A-B приводит к OF=1 A=65535 B= -10

г) A+B приводит к OF=1 A=65535 B = 65535

7 задачи

1. Выполнить вручную расчёт операции, производимой логическим блоком (рис. 4) над знаковыми байтами, найти все разряды результата, значения переносов во всех разрядах, а также определить значения флагов:

а) 01101110‑11101101

б) 10110111+11100111 = 0111001 cf =1 of = 0

в) 11010011‑01011110

2. Выполнить вручную расчёт операции, производимой логическим блоком с 4-разрядным сумматором с параллельным переносом (рис. 5) над знаковыми тетрадами, найти все разряды результата, значения функций генерирования и распространения переносов, значения переносов во всех разрядах, а также определить значения флагов:

а) 0110‑1110

б) 1011+1110 = 1000 cf =1 of =1

в) 1101‑0101

3. Разработать программную модель сумматора с параллельным переносом на языке C#, которая:

* содержит две знаковые тетрады, например:

sbyte x = 0b1001;

sbyte y = 0b0011;

byte z = (byte)((x >> 3) & 1);

Console.WriteLine($"x={x} y={y} \nстарший бит тетрады x = {z}");

* преобразует (при необходимости) тетрады в одномерные массивы битов посредством битовых операций умножения на маску и сдвига;
* воспроизводит битовые операции над разрядами слагаемых и переносами, которые выполняет сумматор с параллельным переносом (функции генерирования и распространения, значения переносов и разрядов суммы);
* выводит на экран функции генерирования и распространения, значения переносов и разрядов суммы;
* вычисляет и выводит на экран значения флагов.

sbyte x = 0b1001;

sbyte y = 0b0011;

int[] xms = new int[5];

int[] yms = new int[5];

int next, s, p, g, c = 0, c1 = 0;

for( byte i = 0; i < 4; i++)

{

next = x & 0x1;

xms[i] = next;

x = Convert.ToSByte(x >> 1);

}

next = 0;

for (byte i = 0; i < 4; i++)

{

next = y & 0x1;

yms[i] = next;

y = Convert.ToSByte(y >> 1);

}

for (byte i = 0; i < 4; i++)

{

Console.Write(xms[i]);

}

Console.WriteLine(" ");

for (byte i = 0; i < 4; i++)

{

Console.Write(yms[i]);

}

Console.WriteLine(" ");

for (byte i = 0; i < 4; i++)

{

if(xms[i] == 1 & yms[i] == 1)

{

s = 0;

p = 0;

g = xms[i] \* yms[i];

c1 = c;

c = g + p \* c1;

Console.WriteLine("C + {0} = {1} ", i, c);

Console.WriteLine("X + {0} = {1} ", i, xms[i]);

Console.WriteLine("Y + {0} = {1} ", i, yms[i]);

Console.WriteLine("S + {0} = {1} ", i, s);

Console.WriteLine("P + {0} = {1} ", i, p);

Console.WriteLine("G + {0} = {1} ", i, g);

}

else if(xms[i] == 1 || yms[i] == 1)

{

s = 1;

p = 1;

g = xms[i] \* yms[i];

c1 = c;

c = g + p \* c1;

Console.WriteLine("C + {0} = {1} ", i, c);

Console.WriteLine("X + {0} = {1} ", i, xms[i]);

Console.WriteLine("Y + {0} = {1} ", i, yms[i]);

Console.WriteLine("S + {0} = {1} ", i, s);

Console.WriteLine("P + {0} = {1} ", i, p);

Console.WriteLine("G + {0} = {1} ", i, g);

}

else if (xms[i] == 0 & yms[i] == 0)

{

s = 0;

p = 0;

g = xms[i] \* yms[i];

c1 = c;

c = g + p \* c1;

Console.WriteLine("C + {0} = {1} ", i, c);

Console.WriteLine("X + {0} = {1} ", i, xms[i]);

Console.WriteLine("Y + {0} = {1} ", i, yms[i]);

Console.WriteLine("S + {0} = {1} ", i, s);

Console.WriteLine("P + {0} = {1} ", i, p);

Console.WriteLine("G + {0} = {1} ", i, g);

}

}

Console.WriteLine("Флаги:");

Console.WriteLine("CF - {0}, ZF - {1}, SF - {2}, PF - {3}",c,0,1,1);

}