

Задача С3. ДВОИЧНА СУМА

Двоичната сума на цели положителни числа се получава чрез сумиране цифрите от двоичното им представяне по модул 2. Полученият резултат е двоичното представяне на двоичната сума.

Например, двоичната сума на числата 8, 15, 12 и 2 е 9:

	2^3	2^2	2^1	2^0	
8 =	1	0	0	0	
15 =	1	1	1	1	
12 =	1	1	0	0	
2 =	0	0	1	0	
	1	0	0	1	= 9

Напишете програма **BSUM**, която намира двоичната сума на зададени числа, като въвежда от първия ред на стандартния вход едно цяло число n ($1 < n \leq 10^6$), а от следващите n реда – числата, чиято двоична сума трябва да се намери. Числата не надхвърлят 10^9 . На стандартния изход се извежда едно цяло число – намерената сума.

ПРИМЕР

Вход	Изход
4	9
8	
15	
12	
2	

Решение:

Зададените ограничения подсказват, че е необходимо да въвеждаме числата едно по едно и последователно да ги добавяме към сумата. Търсената сума получаваме в масив S, елементите на който съдържат цифрите на нейното двоично представяне. За целта зануляваме елементите на масива S, въвеждаме числата едно по едно, преобразуваме всяко в двоичен вид и го добавяме към сумата.

```
#include<iostream.h>
main()
{ int n, x, d, s[30]={0};
  cin>>n;
  int i=0, nd=0;
  // nd съдържа дължината на двоичното представяне на сумата
  // въвеждат се числата
  for (int j=0; j<n; j++)
  {
    cin>>x; i=0;
    //преобразуват се в двоичен вид и всяка от получените двоични
    цифри се добавя към съответния елемент на сумата
    while (x>0)
    {
      d=x%2; x/=2; s[i]=d xor s[i]; i++;
    }
    //обновява се дължината на получената сума
    if (i-1>nd)nd=i-1;
```

```

    }
    //двоичната сума се преобразува в десетичен вид и се извежда
    x=0;
    for (int j=nd;j>=0; j--) x=x*2+s[j];
    cout<<x<<endl;
}

```

Решение с използване на побитови операции:

По условие двоичната сума се получава чрез събиране на цифрите от двоичното представяне по модул 2. Това позволява да използваме операция хог за намиране на двоичната сума. Първото от числата въвеждаме директно в s, където ще натрупваме сумата. Останалите числа въвеждаме последователно в цикъл и ги добавяме към сумата чрез хог. Така получаваме сумата в s, която в края извеждаме.

```

#include<iostream.h>
main()
{ int s, x, n;
  cin>>n;
  cin>>s;
  for (int i=1;i<n;i++)
  {
    cin>>x; s=s xor x;
  }
  cout<<s<<endl;
}

```

Задача D1. САМОЛЕТ

Самолет излита в K часа и M минути и пристига в L часа и N минути. Да се напише програма **PLANE**, която намира колко часа и колко минути е летял самолетът и кое негово време (излитането или кацането) е по-напред в денонощието си. Продължителността на полета е по-малка от 24 часа.

Излитането и кацането са в рамките на една и съща часова зона.

Вход:

Четири цели числа на един ред разделени с интервал K M L N ($0 \leq K, L \leq 23, 0 \leq M, N \leq 59$).

Изход: Две цели числа за часове и минути за продължителността на полета на първия ред и "I" или "K" на втория ред в зависимост, от това кое е по-напред в денонощието си – излитането или кацането.

Примери:

	Пример 1	Пример 2	Пример 3
Вход	1 20 4 40	20 30 2 5	22 0 2 0
Изход	3 20 I	5 35 K	4 0 K

Решение:

При решаването на тази задача е удобно часовете и минутите да се превърнат в минути – по този начин се отчита само в една мярна единица времето изтекло от началото на съответното денонощие до излитането (изчислено е в променливата **t1**) и до кацането (изчислено е в променливата **t2**). Проверката на това кое е по-напред в денонощието след това е лесна – ако **t1<t2**, то тъй като полетът има продължителност по-малка от 24 часа те са в едно и също денонощие, а в противен случай в различни денонощия.