

```

    }
    //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 |
|--------------|-----------|-----------|----------|
| <b>Вход</b>  | 1 20 4 40 | 20 30 2 5 | 22 0 2 0 |
| <b>Изход</b> | 3 20<br>I | 5 35<br>K | 4 0<br>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.

```

#include <iostream>
using namespace std;
int main(){
    int K, M, L, N, t1,t2, h, m;
    cin >>K >>M >>L >>N;
    t1=K*60+M;
    t2=L*60+N;
    if () {
        h=(t2-t1)/60;
        m=(t2-t1)%60;
        cout << h<<" "<<m<<endl<<"I\n";}
    else {
        h=(24*60+t2-t1)/60;
        m=(24*60+t2-t1)%60;
        cout << h<<" "<<m<<endl<<"K\n";}
    return 0;
}

```

## Задача D2. КАРТИНКА

Иванчо сега се учи да чете и пише на английски. По време на пътуване измислил една игра – преписва една дума от речника си на лист само с малки букви и след това я прави на картинка.

- За някои думи постъпва така: на всеки следващ ред преписва отново думата, но без последната буква от предходният ред и това се повтаря докато свърши думата. Например за “book”, се получава:

```

book
boo
bo
b

```

- За други – на следващия ред преписва горната дума, но без първата и последната буква, като запазва местоположението на буквите една под друга. И това се повтаря докато остане 1 или 2 букви, а после ги добавя постепенно отново. Например за “alabala”, се получава:

```

alabala
 labal
  aba
   b
  aba
 labal
alabala

```

Един ден той открил, че пише думите картинки винаги по втория начин, когато те могат да се четат по един и същ начин, от ляво на дясно и от дясно наляво. Кака му (от D група), нали си е винаги всезнайка, му обяснила, че такива думи се наричат “палиндроми” (Ух – отвратително е колко много знае :() )

Напишете програма **HOURLASS**, която за дадена дума извежда картинката, която би нарисувал Иванчо.

**Вход:** Дума имаща не повече от 20 символа. Всички символи са английски букви.

**Изход:** Картинката, която би написал Иванчо.