

```

maxN=minN=0;
for ( i=0; i<5; i++)
{
    minN=minN*10+a[i];
    maxN=maxN*10+a[4-i];
}

```

6. Извежда се разликата между максималното и минималното число:

```

cout<<maxN-minN<<endl;

```

Ето окончателния вид на програмата от втория начин:

```

#include<iostream.h>
void main ()
{
    long N, maxN, minN;
    int a[5], i, j;
    cin >> N;
    for ( i=0; i<5; i++) {a[i]=N%10; N/=10; }
    for ( i=0; i<4; i++)
        for (j=0; j<4; j++)
            if (a[j]>a[j+1]) { int z=a[j]; a[j]=a[j+1]; a[j+1]=z; }
    maxN = minN = 0;
    for ( i=0; i<5; i++)
    {
        minN=minN*10+a[i];
        maxN=maxN*10+a[4-i];
    }
    cout<<maxN-minN<<endl;
}

```

Задача E2. Бонбонки

Умко (нашият познайник от училище “Програмистите на България”) имал странно хоби. Той имал четири кристални купички, в които грижливо събирал шоколадови бонбонки. Странното било това, че той не обичал да си похапва от тях, а само да си ги гледа разпределени по равен брой в своите купички. Във всяка от тях той държал по 10 бонбонки. Това, че нашето умно момче не обичало шоколадовите бонбонки съвсем не означавало, че и неговата сестра не обичала да си похапва сладко, сладко от тях. Тя само изчаквала Умко да отиде на училище и с огромно удоволствие си похапвала. Така всеки ден след като се връщал от училище, той виждал, че любимите му кристални купички вече не са пълни с по 10 бонбонки във всяка и тичал до магазина, за да набави необходимите бонбонки. След известно време Умко сменил тактиката. Вместо всеки ден да ходи до магазина и да си купува бонбонки, просто ги премествал от едната купичка в другата, за да станат пак по равен брой в четирите купички. В случаите, когато това не било възможно, Умко все пак тичал до магазина.

Тъй като му омръзнало всеки път да пресмята колко бонбони трябва да размести и да купи, нашият приятел ви моли да напишете програма **BONBONKI.EXE**, която прочита от клавиатурата последователно броя на бонбонките във всяка една от четирите купички и ако е възможно те да се разпределят по равно, извежда на екрана от коя купичка колко бонбонки се изваждат или добавят. Ако бонбонките се изваждат от купичката, пред броя им се поставя знака “-“, а ако те се добавят – знака “+”. Ако броя на бонбонките в съответната купичка остава непроменен, се извежда числото 0 без знак. Ако не е възможно бонбонките да се разпределят по равно в четирите купички, програмата извежда на екрана само броя на бонбонките, които Умко трябва да купи от магазина, за да поправи разпределението им.

Примерен вход:

2 3 4 6

Примерен изход:

1

Примерен вход:

4 9 8 7

Примерен изход:

+3 -2 -1 0

Решение:

За решението на задачата е необходимо да се намери общия брой бонбони във всички купички и да се намери остатък при деление на 4 на този брой. Ако този остатък е 0, то резултатът при деление на общия брой на 4 е броя на бонбонките, който трябва да се получи във всяка купичка. Тогава за всяка купичка трябва да се провери дали трябва да се добавят и ли извадят бонбонки и да се отпечата съответния брой.

Ако остатъкът при деление на общия брой на 4 е различен от нула, се налага да се купят бонбонки от магазина и техният брой е точно допълнението на остатъка до 4.

Ето една примерна програма, която решава горната задача:

```
#include <iostream.h>
void main()
{
    int a,b,c,d,e;
    cin>>a>>b>>c>>d;
    e=(a+b+c+d)%4;
    if(e) cout<<(4-e)<<endl;
    else
    {
        e=(a+b+c+d)/4;
        if(a>e) cout<<"-"<<a-e<<" ";
        else if(a==e) cout<<"0"<<" ";
        else cout<<"+"<<e-a<<" ";
        if(b>e) cout<<"-"<<b-e<<" ";
        else if(b==e) cout<<"0"<<" ";
        else cout<<"+"<<e-b<<" ";
        if(c>e) cout<<"-"<<c-e<<" ";
        else if(c==e) cout<<"0"<<" ";
        else cout<<"+"<<e-c<<" ";
        if(d>e) cout<<"-"<<d-e<<endl;
        else if(d==e) cout<<"0"<<endl;
        else cout<<"+"<<e-d<<endl;
    }
}
```

Задача Е3. Познай цифрата

По време на междучасието в училището в село Каръшко играели следната игра: Подреждат се плочки, на всяка от които е изписана цифра от 0 до 9. Плочките са обърнати с надписа надолу, така че да не се вижда коя е цифрата на нея и се знае, че с тях са изписани числата от 10 до 99 в нарастващ ред (101112131415...979899).

Един от играчите посочва една от плочките, а този който е наред, познава коя цифра е записана на плочката. Печели този играч, който е познал най-много цифри. Нашият добър приятел Умко и този път иска да надхитри приятелите си, но за целта вие трябва да му помогнете като напишете програма **CIFRA.EXE**, която въвежда число k ($1 \leq k \leq 180$) и извежда цифрата изписана на k -тата плочка.