

Окружно такмичење из рачунарства ученика основних школа, 23.5.2021.

5. разред

1.

У свакој кутији коју је Никола добио за имендан, налази се тачно једна картица на којој је записана једна цифра.

Никола жели да састави највећи могућ број од ових цифри (без окретања картица) Напишите програм који учитава цифре написане на картицама и исписује највећи троцифрени број који можемо формирати на описани начин.

Улаз

У једином реду стандардног улаза дате су три цифре, одвојене једним знаком размака. Можете сматрати да бар једна од цифара није нула.

Излаз

На стандардном излазу, програм мора да испише цео број - највећи троцифрени број који се добије искључиво на горе описан начин.

Пример 1

Улаз

6 2 4

Излаз

642

Пример 2

Улаз

0 8 8

Излаз

880

Решење:

Након уноса улазних података, желимо да уредимо малу серију од 3 цифре у нерастућем поретку тако да најпре испишемо на стандардни излаз редом највећу цифру, средњу, најмању цифру.

```
#include<iostream>
using namespace std;
```

```
int main()
{
    int a,b, c;
    cin >> a >> b >> c;

    if(c>b) swap(c,b);
    if(b>a) swap(b,a);
    if(c>b) swap(c,b);

    cout << a << b << c << endl;
}
```

Можемо задатак решити и поштујући правило о формирању вредности троцифреног броја у бројевном систему са основом 10. Након уноса улазних података, желимо да одредимо највећу и најмању цифру, док ћемо средњи цифру одредити тако што од збира улазних цифара одузмемо редом највећу и најмању цифру.

```
a,b,c=map(int,input().split())
najv=max(a,b,c)
najm=min(a,b,c)
sre=a+b+c-najv-najm
print(najv*100+sre*10+najm)
```

Овај задатак припада по математичком моделу градиву 3. разреда основне школе и представља један од уводних задатака на курсевима програмирања.

Као такав, појављује се у уводним поглављима *Методичке збирке задатака из основа програмирања* (Највећи од три,...).

Задатак се више пута појављивао на такмичењима у сличном облику и његова решења су представљена у нашим Билтенима које објављујемо на сајту ДМС-а.

(нпр. Општинско 2017, 2018, квалификације 2021,...

<https://dms.rs/wp-content/uploads/2017/03/Opstinsko2017Bilten.pdf>

<https://dms.rs/wp-content/uploads/2018/06/BiltenOpstinsko2018.pdf>)

2.

На екипном програмерском такмичењу дозвољено је учешће само мешовитих тимова које чини 3 такмичара (1 дечак и 2 девојчице, или 1 девојчица и 2 дечака). У школи међу ученицима постоји n девојчица и m дечака. Напишите програм који одређује максималан број тимова који може да се састави од ученика ове школе. Нека вас не чуди што ова школа може имати баш велики број ученика.

Улаз

У првој линији стандардног улаза дата су два природна броја n и m ($1 \leq n, m \leq 1000000000000$).

Излаз

На стандардном излазу приказати један цео број једнак максималном број мешовитих тимова који могу учествовати на такмичењу.

Пример 1

Улаз

6 7

Излаз

4

Појашњење:

Од 6 девојчице и 7 дечака могу да се саставе 3 мешовита тима (1 девојчица + 2 дечака) и још 1 мешовит тим (2 девојчице + 1 дечак). На сличан начин могу да се саставе и 2 мешовита тима (2 девојчица + 1 дечак) и још 2 мешовита тима (1 девојчица + 2 дечака).

Пример 2

Улаз

48 90

Потпуно решење може се добити анализом основних случаја. Без смањења општости, ако $N > M$, разменићемо вредности за N, M , те сматрамо да важи $M \geq N$.

Ако $N < M/2$, могуће је направити само N тимова.

Можемо да саставимо $M-N$ тимова, кад изједначимо број учесника у два тима у формацији $(1 + 2)$ и $(2 + 1)$.

Када у свакој групи остане мање од 3 ученика (остатак при дељењу са 3, тј. може да се провери број престалих дечака и девојчица) и да се направи још један тим.

```
# include <iostream>
# include <algorithm>
using namespace std;

int main()
{
    long long n,m,br=0;
    cin>>n>>m;

    if(n>m) swap(n,m);
    int x=n;

    br=br+min(m/2,n);
    if(m%2==1 and n-m/2>2) br++;
    if((n+m)/3<=n) br=(n+m)/3;

    cout<<br<<endl;
    return 0;
}
```

Делимична решења могу се добити анализом минималних и максималних броја могућих група узимајући у обзир да највећи број могућих група није већи од количника укупног броја такмичара и броја 3. Ако број дечака или број девојчица је такав да је мањи од највећег могућег броја група, онда се групе формирају тако да у претпоставимо да у њима је по један припадник бројчано мање групе (дечака или девојчица).

```
n,m=map(int,input().split())
ukupno=n+m
resenje=ukupno//3
if (n<resenje) or (m<resenje):
    print(min(n,m))
else:
    print(resenje)
```

3.

Нада воли да пребројава серије бројева исте парности. Помозите Нади да у датом низу преброји све поднизове узастопних елемената чији су сви елементи исте парности.

Улаз

У првој линији стандардног улаза дат је природан броја n ($1 \leq n \leq 500000$).

У наредној линији дато је n природних бројева који представљају чланове низа. Сваки члан низа није већи од 1 000 000.

Излаз

На стандардном излазу приказати један цео број једнак траженом броју поднизова.

Пример 1

Улаз

5

9 11 13 10 12

Излаз

9

Појашњење:

То су поднизови 9, 9 11, 9 11 13, 11, 11 13, 13, 10, 10 12 и 12.

Решење:

Задатак је креиран по угледу на задатак из *Методичке збирке задатака из алгоритмике* (Број сегмената исте парности)

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
    int n;  
    cin >> n;  
    int prethodni;  
    cin >> prethodni;  
    unsigned long long brojac = 1;  
    int duzina = 1;  
    for (int i = 1; i < n; i++) {  
        int tekuci;  
        cin >> tekuci;  
        if (prethodni % 2 != tekuci % 2)  
            duzina = 0;  
        brojac += ++duzina;  
        prethodni = tekuci;  
    }  
    cout << brojac << endl;  
    return 0;
```

Алгоритам грубе силе подразумева да се провере сви сегменти дужине која је већа или једнака 1.

Ако редом пролазимо по елементима низа и за сваки сегмент увек изнова проверавамо да ли су сви елементи исте парности (на пример, тако што проверавамо да ли је сваки елемент исте парности као њему претходни) добијамо алгоритам који више пута пролази кроз низ (и више пута проверава) парност истог члана низа.

Зато још при читавању сваког члана низа редом делимо учитани део на најдуже могуће сегменте у којима су сви елементи исте парности и памтимо дужину текућег сегмента елемената исте парности. Ако је следећи елемент исте парности као претходни, само продужавамо најдужи могући сегмент (увећавањем дужине), а у супротном обрађујемо управо завршени сегмент и започињемо нови (он је једночлан и чини га само текући елемент који смо читали).

```

using System;
class Program
{
    static void Main(string[] args)
    {
        int n = int.Parse(Console.ReadLine());
        int prethodni = int.Parse(Console.ReadLine());
        int brojSegmenata = 1;
        int duzina = 1;
        for (int i = 1; i < n; i++) {
            int tekuci = int.Parse(Console.ReadLine());
            if (prethodni % 2 != tekuci % 2)
                duzina = 0;
            brojSegmenata += ++duzina;
            prethodni = tekuci;
        }
        Console.WriteLine(brojSegmenata);
    }
}

```

4.

На дугачкој траци записани су један за другим сви цели бројеви од 1 до n. Сваки од ових бројева записан је тачно једном и ниједан број се не понавља. Бројеви не морају бити записани у растућем поретку. Између свака два броја постоји празно мести на траци. Желимо да разрежемо траку и добијемо поднизовете растућих узастопних природних бројева (без окретања). Можемо извршити пресецање траке на позицији празног места. Није дозвољено пресећи траку између цифара једног броја. Напишите програм који проналази колико најмање резова морамо направити тако да бројеви у добијеним деловима буду распоређени на наведени начин.

Улаз

У првој линији стандардног улаза дат је природан број n ($1 \leq n \leq 999999$).

У следећој линији стандардног улаза дато је n целих бројева оним редоследом којим су записани на траци. Свака два суседна броја су одвојена размаком.

Излаз

На стандардном излазу приказати два цела броја одвојена тачно једним размаком: потребан број направљених резова и број чланова низа који се налазе у најдужем делу траке добијеном након резања.

Пример 1

Улаз

14

12 13 1 2 4 5 6 7 10 11 3 8 9 14

Излаз

6 4

Појашњење:

Позиције разрезавања на траци ћемо означити симболом #:

12 13 # 1 2 # 4 5 6 7 # 10 11 # 3 # 8 9 # 14

Учимо да постоји 6 резова и да најдужи део траке након резова садржи бројеве 4 5 6 7, тј. број чланова тог подниза је 4.

Пример 2

Улаз

10

1 2 3 4 5 6 7 8 9 10

Изназ

0 10

Решење:

Ако погледамо објашњење примера 1, јасно се уочава идеја за решавање задатка. Дакле, улазна колекција бројева се састоји од делова (сегмената, ланаца) узастопних бројева. У сваком случају, потребно је потражити такве сегменте и њихове граничнике (у објашњењу означене знаком #), и сваки пут када је ланац прекинут забележити дужину до тада најдугег ланца и бројач дужине ланца поставити на вредност 1.

```
#include<iostream>
using namespace std;
int n;
int main()
{
    cin >> n;
    int rez=0, duzina=1, najvecauzina=1;
    int a;
    cin >> a;

    for(int i=2; i<=n; i++)
    {
        int b;
        cin >> b;
        //ako je prekinut lanac uzaostopno ucitanih brojeva
        if(b!=a+1)
            {rez++;
            if(najvecaduzina<duzina)najvecaduzina=duzina;
            duzina=1;
            }
        else duzina++;
        a=b;
    }

    if(najvecaduzina<duzina)najvecaduzina=duzina;
    cout << rez << " " << najvecauzina << endl;
}
```

6. разред

1.

Јелисавета везе гоблен за своју другарицу и прати мустре на којима су приказане све цифре изузев цифре нула.

Јелисавета жели да састави најмањи могућ број од ових цифри (без окретања мустре). Напишите програм који читава цифре приказане на мустрима и испишује најмањи троцифрени број који Јелисавета може да изведе.

Улаз

У једином реду стандардног улаза дате су три цифре, одвојене једним знаком размака.

Израз

На стандардном излазу, програм мора да испише цео број - најмањи троцифрени број који ће се приказати на гоблену на горе описан начин.

Пример 1

Улаз

9 8 4

Израз

489

Пример 2

Улаз

8 8 8

Израз

888

Решење:

Након уноса улазних података, желимо да уредимо малу серију од 3 цифре у нерастућем поретку тако да најпре испишемо на стандардни излаз редом намању цифру, средњу, највећу цифру.

```
#include<iostream>
using namespace std;
```

```
int main()
{
    int a,b, c;
    cin >> a >> b >> c;
    if(c<b) swap(c,b);
    if(b<a) swap(b,a);
    if(c<b) swap(c,b);
    cout << a << b << c << endl;
}
```

Можемо задатак решити и поштујући правило о формирању вредности троцифреног броја у бројевном систему са основом 10. Након уноса улазних података, желимо да одредимо најмању и највећу цифру, док ћемо средњу цифру одредити тако што од збира улазних цифара одузмемо редом највећу и најмању цифру.

```
a,b,c=map(int,input().split())
najv=max(a,b,c)
najm=min(a,b,c)
sre=a+b+c-najv-najm
print(najm*100+sre*10+najv)
```

Овај задатак припада по математичком моделу градиву 3. разреда основне школе и представља један од уводних задатака на курсевима програмирања.

Као такав, појављује се у уводним поглављима *Методичке збирка задатака из основа програмирања* (Највећи од три,...).

Задатак се више пута појављивао на такмичењима у сличном облику и његова решења су представљена у нашим Билтенима које објављујемо на сајту ДМС-а.

(нпр. Општинско 2017, 2018, квалификације 2021,...

<https://dms.rs/wp-content/uploads/2017/03/Opstinsko2017Bilten.pdf>

<https://dms.rs/wp-content/uploads/2018/06/BiltenOpstinsko2018.pdf>)

2.

Пера је током Округног такмичења из рачунарства предао први задатак на аутоматском оцењивачу више пута. Број поена које Пера осваја се рачуна тако што се одреди највећи број поена међу свим до тада предатим Периним решењима тог задатка. Напиши програм који одређује колико је поена за први задатак Пера освојио након сваког слања. Немојте да Вас зачуди што први задатак може да вреди много поена и што је Пера послао релативно велики број решења аутоматском оцењивачу.

Улаз

У првој линији стандардног улаза налази се природан број n ($n \leq 50000$). У следећих n линија налазе се редом поени које је Пера освојио за свако појединачно слање (број између 0 и 100000).

Излаз

На стандардном излазу приказати n линија које приказују колико је поена за тај задатак Пера имао након сваког слања.

Пример

Улаз

5

1

2

4

1

5

Излаз

1

2

4

4

5

Решење:

Задатак је креиран по угледу на задатак из *Методичке збирка задатака из основа програмирања* (Најбољи сабмит)

```
#include <iostream>
using namespace std;
int main() {
    ios_base::sync_with_stdio(false); cin.tie(0);
    int n;
    cin >> n;
    int maxPoena = 0;
    for (int i = 0; i < n; i++) {
        int poeni;
        cin >> poeni;
        if (poeni > maxPoena)
            maxPoena = poeni;
        cout << maxPoena << '\n';
    }
    return 0;
}
```

Методом лажне претпоставке прогласимо да највећи број поена које је Пера могао освојити на почетку је једнак 0. Потом примењујемо алгоритам одређивања максимума серије бројева тако што обарамо или потврђујемо претпоставку упоређивањем максимума Периних поена до тада са текућим бројем освојених поена. Али се у сваком кораку исписује текућа вредност максимума.

```
using System;
```

```
class Program
{
    static void Main(string[] args)
    {
        int n = int.Parse(Console.ReadLine());
        int maxPoena = 0;
        for (int i = 0; i < n; i++) {
            int poeni = int.Parse(Console.ReadLine());
            if (poeni > maxPoena)
                maxPoena = poeni;
            Console.WriteLine(maxPoena);
        }
    }
}
```

3.

Нада воли да пребројава серије бројева исте парности. Помозите Нади да у датом низу преброји све поднизове узастопних елемената чији су сви елементи исте парности.

Улаз

У првој линији стандардног улаза дат је природан броја n ($1 \leq n \leq 50000$).

У наредној линији дато је n природних бројева који представљају чланове низа. Сваки члан низа није већи од 1 000 000.

Излаз

На стандардном излазу приказати један цео број једнак траженом броју поднизова.

Пример 1

Улаз

5

9 11 13 10 12

Излаз

9

Појашњење:

То су поднизови 9, 9 11, 9 11 13, 11, 11 13, 13, 10, 10 12 и 12.

Решење:

Задатак је креиран по угледу на задатак из *Методичке збирке задатака из алгоритмике* (Број сегмената исте парности)

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
    int n;  
    cin >> n;  
    int prethodni;  
    cin >> prethodni;  
    unsigned long long brojac = 1;  
    int duzina = 1;  
    for (int i = 1; i < n; i++) {  
        int tekuci;  
        cin >> tekuci;  
        if (prethodni % 2 != tekuci % 2)  
            duzina = 0;  
        brojac += ++duzina;  
        prethodni = tekuci;  
    }  
    cout << brojac << endl;  
    return 0;  
}
```

Алгоритам грубе силе подразумева да се провере сви сегменти дужине која је већа или једнака 1.

Ако редом пролазимо по елементима низа и за сваки сегмент увек изнова проверавамо да ли су сви елементи исте парности (на пример, тако што проверавамо да ли је сваки елемент исте парности као њему претходни) добијамо алгоритам који више пута пролази кроз низ (и више пута проверава) парност истог члана низа.

Зато још при читавању сваког члана низа редом делимо учитани део на најдуже могуће сегменте у којима су сви елементи исте парности и памтимо дужину текућег сегмента елемената исте парности. Ако је следећи елемент исте парности као претходни, само продужавамо најдужи могући сегмент (увећавањем дужине), а у супротном обрађујемо управо завршени сегмент и започињемо нови (он је једночлан и чини га само текући елемент који смо читали).

```
using System;
class Program
{
    static void Main(string[] args)
    {
        int n = int.Parse(Console.ReadLine());
        int prethodni = int.Parse(Console.ReadLine());
        int brojSegmenata = 1;
        int duzina = 1;
        for (int i = 1; i < n; i++) {
            int tekuci = int.Parse(Console.ReadLine());
            if (prethodni % 2 != tekuci % 2)
                duzina = 0;
            brojSegmenata += ++duzina;
            prethodni = tekuci;
        }
        Console.WriteLine(brojSegmenata);
    }
}
```

4.

Дат је низ од n ненегативних различитих целих бројева a_1, a_2, \dots, a_n са вредностима у опсегу 0 до $n - 1$. Лаза жели да изврши операцију размене у којој се бирају два броја из низа и трампе се (њихове позиције у низу се размене). Напишите програм који проналази са колико размена Лаза може улазни низ распоредити у растућем редоследу.

Улаз

У првој линији стандардног улаза налази се природан број n ($3 \leq n \leq 99999$). У следећем реду је дато n чланова низа раздвојених међусобно са по једним размаком.

Излаз

На стандардном излазу приказати један цео број једнак потребном минималном броју размена.

Пример

Улаз

8

7 3 0 1 4 5 6 2

Излаз

3

Појашњење:

Разменимо 7 и 2. Добијамо низ 2 3 0 1 4 5 6 7

Разменимо 0 и 2. Добијамо низ 0 3 2 1 4 5 6 7

Разменимо 1 и 3. Добијамо низ 0 1 2 3 4 5 6 7

Није потребно више од 3 размене, јер смо добили растући низ.

Решење:

```
#include<bits/stdc++.h>
using namespace std;

const int maxn=1e5;
int a[maxn];
vector<int> v;

int main(){
    int n;
    cin>>n;

    for(int i=0;i<n;i++){
        int broj;
        cin>>broj;
        v.push_back(broj);
        a[v[i]]=i;
    }

    int sum=0;
    for(int i=0;i<n;i++){
        if(a[i]!=i){
            swap(v[a[i]], v[i]);
            swap(a[v[a[i]]], a[v[i]]);
            sum++;
        }
    }
    cout<<sum;
    return 0;
}
```


7. разред

1.

Мика је (по казни) током часа на даљину записивао ненегативне целе бројеве 0 до n . Али, Мики прети нова казна, јер није записао тачно један број. Напиши програм који учитава бројеве које је Мика записао и исписује број који је Мика случајно изоставио.

Улаз

У првој линији стандардног улаза налази се природан број n ($1 \leq n \leq 10^9$). У следећој линији налази се горе описани низ бројева. Бројеви су раздвојени са по једним размаком.

Излаз

На стандардном излазу приказати број који је Мика изоставио.

Пример

Улаз

12

11 10 7 4 6 2 5 0 1 9 3 12

Излаз

8

Решење:

Задатак је креиран по угледу на задатак из *Методичке збирка задатака* (Недостајући број)

```
#include <iostream>
using namespace std;
int main() {
    ios_base::sync_with_stdio(false);
    int n;
    cin >> n;
    long long zbir = 0;
    for (int i = 0; i < n; i++) {
        int x; cin >> x;
        zbir += x;
    }
    long long zbir_svih = ((long long)n) * (n+1) / 2;
    int nedostajuci = zbir_svih - zbir;
    cout << nedostajuci << endl;
    return 0;
}
```

Збир свих елемената из скупа $\{0, 1, 2, \dots, n\}$ је $0 + 1 + 2 + \dots + n = \frac{n \times (n+1)}{2}$

(Сетимо се анегдоте из живота малог Гауса из 4. разреда који је брзо сабрао бројеве од 1 до 100)

То је збир елемената који се налазе у низу и недостајућег елемента. Отуда, недостајући елемент је, једнак, разлици између $\frac{n \times (n+1)}{2}$ и збира свих елемената низа.

```

using System;

class Program
{
    static void Main(string[] args)
    {
        int n = int.Parse(Console.ReadLine());
        long zbir = 0;
        string[] str = Console.ReadLine().Split();
        for (int i = 0; i < n; i++)
        {
            int x = int.Parse(str[i]);
            zbir += x;
        }
        long zbir_svih = ((long)n) * (n+1) / 2;
        int nedostajuci = (int)(zbir_svih - zbir);
        Console.WriteLine(nedostajuci);
    }
}

```

2.

Ада је током пандемије имала времена да се бави разним рачунарским проблемима. Она пише редом узастопне природне бројеве од 1 до неког већег броја и тако прави велики низ природних бројева. Затим сваки од бројева у великом низу замени збиром његових цифара и то ради док сви бројеви у низу не постану једноцифрени.

На пример почетна вредност 888. члана низа је број 888. Ада мења тај број вредношћу збира $8 + 8 + 8 = 24$, затим понавља сабирање цифара броја 24 и добија: $2 + 4 = 6$. Ада потом контактира Чеду и задаје му К питања: „Који је збир бројева у новонасталом низу од члана А.-тог члана до В.-тог члана(укључујући А.-ти и В.-ти члан)?

Улаз

У првој линији стандардног улаза налази се природан број K ($1 \leq K \leq 100$) тј. број питања која поставља Ада. У свакој од следећих K линија налази се по једно питање које садржи два позитивна цела броја A_i и B_i , одвојених знаком размака. Оба броја описују i -то питање ($1 \leq A_i \leq B_i < 2^{60}$ за $i=1 \dots K$).

Излаз

На стандардном излазу приказати K линија које приказују одговор на свако Адино питање. Сваки одговор мора бити одштампан у посебном реду и њихов редослед мора одговарати редоследу питања постављених на улазу.

Пример 1

Улаз

1

1 3

Излаз

6

Пример 2

Улаз

2

9 11

100 101

Излаз

12

3

Решење:

```
#include <iostream>
```

```
using namespace std;
```

```
int k;
```

```
int main()
```

```
{
```

```
cin>>k;
```

```
while (k--)
```

```
{
```

```
unsigned long long int a, b;
```

```
cin >> a >> b;
```

```
unsigned long long int raspon = b - a + 1;
```

```
unsigned long long int zbir = (raspon / 9) * 45;
```

```
unsigned long long int i;
```

```
for (i = 0; i < raspon % 9; ++i)
```

```
{
```

```
unsigned long long int dodaj= (a + i) % 9;
```

```
if (dodaj == 0) dodaj = 9;
```

```
zbir += dodaj;
```

```
}
```

```
cout << zbir << endl;;
```

```
}
```

```
return 0;
```

```
}
```

3.

Дат је низ од n ненегативних различитих целих бројева a_1, a_2, \dots, a_n са вредностима у опсегу 0 до $n - 1$. Лаза жели да изврши операцију размене у којој се бирају два броја из низа и трампе се (њихове позиције у низу се размене). Напишите програм који проналази са колико размена Лаза може улазати низ распоредити у растућем редоследу.

Улаз

У првој линији стандардног улаза налази се природан број n ($3 \leq n \leq 99999$). У следећем реду је дато n чланова низа раздвојених међусобно са по једним размаком.

Излаз

На стандардном излазу приказати један цео број једнак потребном минималном броју размена.

Пример

Улаз

8

7 3 0 1 4 5 6 2

Израз

3

Појашњење:

Разменимо 7 и 2. Добијамо низ 2 3 0 1 4 5 6 7

Разменимо 0 и 2. Добијамо низ 0 3 2 1 4 5 6 7

Разменимо 1 и 3. Добијамо низ 0 1 2 3 4 5 6 7

Није потребно више од 3 размене, јер смо добили растући низ.

Решење:

```
#include<bits/stdc++.h>
using namespace std;
```

```
const int maxn=1e5;
int a[maxn];
vector<int> v;
```

```
int main(){
    int n;
    cin>>n;

    for(int i=0;i<n;i++){
        int broj;
        cin>>broj;
        v.push_back(broj);
        a[v[i]]=i;
    }

    int sum=0;
    for(int i=0;i<n;i++){
        if(a[i]!=i){

            swap(v[a[i]], v[i]);
            swap(a[v[a[i]]], a[v[i]]);
            sum++;
        }
    }
    cout<<sum;
    return 0;
}
```

4.

Ленка и Леонардо воле математичке игре. Леонардо је измислио следећи низ бројева 2, 4, 6, 9, 12, 15, 20, 25, ... У том низу сваки члан, почев од другог, једнак је збиру од претходног члана и највећег простог делиоца претходног члана.

Леонардо задаје Ленки број n , а Ленка мора брзо да погоди n -ти члан низа који је Леонардо смислио.

Улаз

У првој линији стандардног улаза налази се природан број n ($1 \leq n \leq 8 \times 10^4$).

Излаз

На стандардном излазу приказати један број, тражени члан низа.

Пример 1

Улаз

3

Излаз

6

Пример 2

Улаз

11

Излаз

42

```
#include<bits/stdc++.h>
using namespace std;
```

```
const int limitsN=80000;
int maxPrimeDiv(int n) {
    if (n == 0) return 2;
    for (int i = 2; i*i <= n; i++) {
        if (n%i == 0) {
            n /= i;
            i--;
        }
    }
    return n;
}
```

```
int ukupno(int n) {
    int a = 2;
    for (int i = 1; i < n; i++) {
        a += maxPrimeDiv(a);
    }
    return a;
}
```

```
void calc() {
    for (int i = 0; i <= limitsN; i+= 1000) { //perioda
        cout << ukupno(i) << ",";
    }
}
```

```
//
```

```
// vremenska ogranicenja u zadatku dozvoljavaju rad i bez predizracunavanja
int precalc[] = {0,
250997,1001985,2252997,4003997,6254997,9006001,12256997,16008001,20258965,25009997,30
261001,36011965,42262901,49014001,56264997,64015937,72267001,81018001,90268985,10001
9965,110271001,121021997,132272997,144023965,156274997,169026001,182276997,196027997
,210278997,225029857,240280985,256032001,272282937,289033937,306284985,324035857,342
286997,361038001,380289001,400039985,420290965,441042001,462292997,484043997,5062950
01,529045997,552296985,576048001,600298997,625049857,650300517,676051997,702303001,7
29053937,756304937,784056001,812306997,841057937,870309001,900059901,930310985,96106
1857,992312901,1024063997,1056314997,1089065997,1122316997,1156067985,1190319001,122
5069601,1260320965,1296071997,1332322985,1369073997,1406325001,1444075985,148232700
1,1521077937,1560328997,1600079937};
```

```
int main() {
    int n; cin >> n;
    int a = precalc[n/1000];
    n %= 1000;
    for (int i = 0; i < n; i++) {
        a += maxPrimeDiv(a);
    }

    cout << a << endl;

    return 0;
}
```

8. разред

1.

Ленка и Леонардо воле математичке игре. Леонардо је измислио следећи низ бројева 2, 4, 6, 9, 12, 15, 20, 25, ... У том низу сваки члан, почев од другог, једнак је збиру од претходног члана и највећег простог делиоца претходног члана.

Леонардо задаје Ленки број n , а Ленка мора брзо да погоди n -ти члан низа који је Леонардо смислио.

Улаз

У првој линији стандардног улаза налази се природан број n ($1 \leq n \leq 8 \times 10^4$).

Излаз

На стандардном излазу приказати један број, тражени члан низа.

Пример 1

Улаз

3

Излаз

6

Пример 2

Улаз

11

Излаз

42

Решење

Видети исти задатак за 7. разред

2.

Мика је (по казни) током часа на даљину записивао ненегативне целе бројеве 0 до n . Али, Мики прети нова казна, јер није записао тачно један број. Напиши програм који читава бројеве које је Мика записао и исписује број који је Мика случајно изоставио.

Улаз

У првој линији стандардног улаза налази се природан број n ($1 \leq n \leq 10^9$). У следећој линији налази се горе описани низ бројева. Бројеви су раздвојени са по једним размаком.

Излаз

На стандардном излазу приказати број који је Мика изоставио.

Пример

Улаз

12

11 10 7 4 6 2 5 0 1 9 3 12

Излаз

8

Решење:

Задатак је креиран по угледу на задатак из *Методичке збирка задатака* (Недостајући број)

```
#include <iostream>
using namespace std;
int main() {
```

```

ios_base::sync_with_stdio(false);
int n;
cin >> n;
long long zbir = 0;
for (int i = 0; i < n; i++) {
    int x; cin >> x;
    zbir += x;
}
long long zbir_svih = ((long long)n) * (n+1) / 2;
int nedostajuci = zbir_svih - zbir;
cout << nedostajuci << endl;
return 0;
}

```

Збир свих елемената из скупа $\{0, 1, 2, \dots, n\}$ је $0 + 1 + 2 + \dots + n = \frac{n \times (n+1)}{2}$

(Сетимо се анегдоте из живота малог Гауса из 4. разреда који је брзо сабрао бројеве од 1 до 100)

То је збир елемената који се налазе у низу и недостајућег елемента. Отуда, недостајући елемент је, једнак, разлици између $\frac{n \times (n+1)}{2}$ и збира свих елемената низа.

```
using System;
```

```

class Program
{
    static void Main(string[] args)
    {
        int n = int.Parse(Console.ReadLine());
        long zbir = 0;
        string[] str = Console.ReadLine().Split();
        for (int i = 0; i < n; i++)
        {
            int x = int.Parse(str[i]);
            zbir += x;
        }
        long zbir_svih = ((long)n) * (n+1) / 2;
        int nedostajuci = (int)(zbir_svih - zbir);
        Console.WriteLine(nedostajuci);
    }
}

```

3.

Ада је током пандемије имала времена да се бави разним рачунарским проблемима. Она пише редом узастопне природне бројеве од 1 до неког већег броја и тако прави велики низ природних бројева. Затим сваки од бројева у великом низу замени збиром његових цифара и то ради док сви бројеви у низу не постану једноцифрени.

На пример почетна вредност 888. члана низа је број 888. Ада мења тај број вредношћу збира $8 + 8 + 8 = 24$, затим понавља сабирање цифара броја 24 и добија: $2 + 4 = 6$. Ада потом

контактира Чеду и задаје му K питања: „Који је збир бројева у новонасталом низу од члана A_i -тог члана до B_i -тог члана(укључујући A_i -ти и B_i -ти члан)?

Улаз

У првој линији стандардног улаза налази се природан број K ($1 \leq K \leq 100$) тј. број питања која поставља Ада. У свакој од следећих K линија налази се по једно питање које садржи два позитивна цела броја A_i и B_i , одвојених знаком размака. Оба броја описују i -то питање ($1 \leq A_i \leq B_i < 2^{60}$ за $i=1 \dots K$).

Излаз

На стандардном излазу приказати K линија које приказују одговор на свако Адино питање. Сваки одговор мора бити одштампан у посебном реду и њихов редослед мора одговарати редоследу питања постављених на улазу.

Пример 1

Улаз

1

1 3

Излаз

6

Пример 2

Улаз

2

9 11

100 101

Излаз

12

3

Решење

Видети исти задатак за 7. разред

4.

Дато је n целих позитивних бројева. Светлана жели да изврши спајање једног броја са другим надовезивањем тако да се добије највећи могући број.

Улаз

У првој линији стандардног улаза налази се природан број n ($1 \leq n \leq 105$). У следећој линији налази се n целих бројева x ($0 \leq x \leq 109$). Бројеви су раздвојени са по једним размаком.

Излаз

На стандардном излазу приказати један број, тражени највећи број који се добија надовезивањем.

Пример

Улаз

4

45 88 7 9

Излаз

988745

```
#include<bits/stdc++.h>
using namespace std;

string a[100000];

bool uporedi(const string& a, const string& b) {
    for (int i = 0; i < a.size()+b.size(); i++) {
        char aa = i<a.size()?a[i]:b[i-a.size()];
        char bb = i<b.size()?b[i]:a[i-b.size()];
        if (aa != bb) return aa > bb;
    }
    return false;
}

int main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);

    int n; cin >> n;
    for (int i = 0; i < n; i++) cin >> a[i];

    sort(a, a+n, uporedi);
    for (int i = 0; i < n; i++) cout << a[i];
    cout << endl;

    return 0;
}
```