

МАТЕМАТИЧКИ ФАКУЛТЕТ У БЕОГРАДУ
- ИНФОРМАТИКА -

Увод у програмирање

- РЕШЕЊА ДОМАЋИХ ЗАДАТАКА -

Аутор

Никола Маринковић

52/2025

Асистент

Луција Миличић

Први семестар, 2025./2026.

Садржај

1		4
1.1		4
1.2		5
2		6
2.1		6
2.2		7
3		8
3.1		8
3.2		9
4		10
4.1		10
4.2		12
5		14
5.1		14
5.2		15
6		16
6.1		16
6.2		18
7		19
7.1		19
7.2		21
8		23
8.1		23
8.2		24
9		25
9.1		25
9.2		27
10		29
10.1		29
10.2		31

Пролог

Документ представља решења свих домаћих задатака датих током школске 2026./2027. године на курсу Увод у програмирање. Документ **није** направљен у сарадњи са предавачима и не представља званични материјал.

Када се у тексту спомиње „крај улаза“, мисли се на пречицу Control+D. Ова пречица има обичај да не функционише на Windows системима, стога је најлакше прећи на линукс.

Понекад ће се програм окончати пре него што завршите са уносом улаза (на пример када је улаз неисправан). Ово је нормално, и ако ваш код поседује идентично понашање, нећете губити поене на тестеру.

Уколико уочите грешке или имате било какве замерке, обратите ми се путем мејла mi25052@alas.matf.bg.ac.rs или путем discord сервера.

I

1.1

Написати програм који читава дужину странице правилног шестоугла изражену у центриметрима и испишује његову површину, изражену у центиметрима квадратним, заокружену на две децимале.

Са стандардног улаза се читава један реалан број који представља дужину странице правилног шестоугла. На стандардни излаз исписати површину шестоугла, заокружену на две децимале.

Улаз:

6.2

Излаз:

99.87

Улаз:

4.75

Излаз:

58.62

Улаз:

0.229

Излаз:

0.14

Улаз:

5

Излаз:

64.95

Решење

```
1 #include <iostream>
2 #include <cmath>
3
4 #include <iomanip>
5
6 using namespace std;
7
8 int main() {
9     double a;
10    cin >> a;
11    if(a<=0) {
12        cout << "-1" << endl;
13        return -1;
14    }
15    cout << fixed << setprecision(2) << 3*sqrt(3)*a*a/2.0 << endl;
16 }
```

1.2

Спортски клуб жели својим навијачима да поклони новогодишње шалове. Шалове је могуће наручити у пакетима од по 100. Клуб има податке о посећености претходних 5 утакмица. На основу тих података, клуб жели да израчуна просечан број посетилаца и под претпоставком да ће толико људи доћи на следећу утакмицу, наручи довољно пакета да сваки навијач добије по шал.

Са стандардног улаза се учитава пет целих бројева који представљају посећеност претходних пет утакмица. На стандардни излаз исписати број пакета који клуб треба да наручи.

Улаз:	Улаз:	Улаз:	Улаз:
1200	150	250	25
1350	176	476	47
1300	202	802	80
1400	123	123	123
1250	182	682	22
Излаз:	Излаз:	Излаз:	Излаз:
13	2	5	1

Решење

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int uk = 0;
7     for(int i = 0; i < 5; i++) {
8         int p;
9         cin >> p;
10        if(p<0) {
11            cout << "-1" << endl;
12            return -1;
13        }
14        uk += p;
15    }
16
17    cout << ((uk/5-1)/100+1) << endl;
18 }
```

II

2.1

Ланац кафића нуди акцију којом купац за купљених k кафа добија наредну кафу бесплатно. Написати програм који омогућава купцу да израчуна колико још кафа треба да купи до прве следеће бесплатне, ако је познат укупан број кафа које је до сад купио у том кафићу (n).

Са стандардног улаза учитавају се редом цели бројеви k и n . На стандардни излаз исписати број кафа до наредне бесплатне. Подразумевати да је улаз исправано задат.

Улаз:

9

4

Излаз:

5

Улаз:

9

13

Излаз:

5

Улаз:

9

9

Излаз:

9

Улаз:

9

18

Излаз:

9

Решење

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int k, n;
7     cin >> k >> n;
8     cout << k-(n%k) << endl;
9 }
```

2.2

Написати програм за шифровање четвороцифрених бројева по принципу описаном у наставку. Цифре броја се прво запишу у обрнутом редоследу, а затим се са десне стране допише контролна цифра. Контролна цифра се добија као последња цифра збира свих цифара задатог броја.

Са стандардног улаза учитава се један позитиван четвороцифрен број. На стандардни излаз исписати број добијен претходно описаним шифровањем.

Улаз:

1234

Излаз:

43210

Улаз:

1111

Излаз:

11114

Улаз:

3456

Излаз:

65438

Улаз:

1212

Излаз:

21216

Решење

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int b;
7     cin >> b;
8
9     if(b<1000||b>9999) {
10         cout << "-1" << endl;
11         return -1;
12     }
13
14     int n = 4;
15     int zb = 0;
16
17     do {
18         int c = b%10;
19         zb += c;
20         cout << c;
21         b/=10;
22     } while(b!=0);
23
24     cout << (zb%10) << endl;
25 }
```

III

3.1

Написати програм који на основу дужина страница проверава да ли је троугао правоугли.

Са стандардног улаза се учитавају се три цела броја која представљају дужине страница неког троугла. На стандардни излаз исписати да ако је дати троугао правоугли, а не иначе. Подразумевати да је улаз исправно задат.

Улаз:

6
8
10

Излаз:

da

Улаз:

15
15
15

Излаз:

ne

Улаз:

12
20
16

Излаз:

da

Улаз:

13
12
5

Излаз:

da

Решење

```
1 #include <iostream>
2
3 using namespace std;
4
5 void razmena(int &a, int &b) {
6     int temp = a; a = b; b = temp;
7 }
8
9 int main() {
10     int a, b, c;
11     cin >> a >> b >> c;
12     if(a<=0 || b<=0 || c<=0) {
13         cout << "ne" << endl;
14         return -1;
15     }
16
17     if(a>b) razmena(a,b);
18     if(b>c) razmena(b,c);
19     if(a>b) razmena(a,b);
20
21     cout << ((a*a+b*b==c*c)? "da" : "ne") << endl;
22 }
```

3.2

Написати програм који на основу године одређује ком историјском раздобљу дата година припада ако је познато да праисторија траје до 3500. године пре нове ере (не укључујући 3500. годину), стари век од 3500 године п.н.е. (укључујући и њу) до 476. године н.е. (не укључујући њу), средњи век од 476. године н.е. (укључујући и њу) до 1492. (без ње), нови век од 1492. године (укључујући и њу) до 1918. године (не укључујући и њу), а савремено доба од 1918. (укључујући и њу) до данашњег дана.

Са стандардног улаза уноси се година (цео број). На стандардни излаз исписати ком добу припада. Године пре нове ере записане су као негативни бројеви.

Улаз:
-3501

Излаз:
praistorija

Улаз:
-3500

Излаз:
stari vek

Улаз:
476

Излаз:
srednji vek

Улаз:
1500

Излаз:
novi vek

Улаз:
2019

Излаз:
savremeno doba

Решење

```

1 #include <iostream>
2
3 using namespace std;
4
5 string selektor(int g) {
6     if(g<-3500) return "praistorija";
7     if(g<476) return "stari vek";
8     if(g<1492) return "srednji vek";
9     if(g<1918) return "novi vek";
10    return "savremeno doba";
11 }
12
13 int main() {
14     int g;
15     cin >> g;
16     cout << selektor(g) << endl;
17 }
```

IV

4.1

Студенти на предмету Увод у програмирање имају домаће задатке који укупно вреде 10 поена. Поред тога, на испиту имају 5 задатака од којих сваки вреди 11 поена. Студенту се као резултат практичног дела испита рачуна боља од следеће две опције:

- $broj_poena_zadaci_sa_ispita$
- $broj_poena_domaci + \frac{broj_poena_zadaci_sa_ispita \cdot 45}{55}$

Са стандардног улаза се учитава један позитиван цео број који представља број поена које је студент остварио на домаћим задацима. Након тога се учитава 5 целих бројева који представљају поене које је студент остварио на задацима са испита. На стандардни излаз исписати број поена који је студент остварио на практичном делу испита.

Напомена: Уколико број поена на практичном делу испита није цео број, резултат заокружити **навише**.

Улаз: 7 9 10 4 7 2 Излаз: 34	Улаз: 4 9 10 4 7 2 Излаз: 32	Улаз: 10 11 9 11 11 11 Излаз: 54	Улаз: 10 5 0 5 0 5 Излаз: 23
--	--	--	--

Решење

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main() {
7     int d;
8     cin >> d;
9     if(d<0||d>10) {
10         cout << "-1" << endl;
11         return -1;
12     }
13
14     int ispit = 0;
15     for(int i = 0; i < 5; i++) {
16         int z;
17         cin >> z;
18         if(z<0||z>11) {
19             cout << "-1" << endl;
20             return -1;
21         }
22         ispit+=z;
23     }
24
25     int komb = d + ((int)ceil((ispit * 45)/55.0));
26     cout << ((ispit > komb)?ispit:komb) << endl;
27 }
```

4.2

На сајму књига важе следећа снижења:

- Ако купац купи барем 10 књига, добија 30% попушта на укупну цену књига.
- Иначе, ако купац купи барем 7 књига, добија 20% попушта на укупну цену књига.
- Иначе, ако купац купи барем 3 књиге, добија 10% попушта на укупну цену књига.

Додатно, неки купци су добили наградне купоне, па за њих важе додатни попусти:

- Ако купац има наградни купон, а број купљених књига је паран, на снижену цену добија још 500 динара попушта.
- Ако купац има наградни купон, а број купљених књига је непаран, на снижену цену добија још 700 динара попушта.

Са стандардног улаза се учитава број купљених књига (позитиван цео број), укупна цена књига без попушта (позитиван цео број) и ознака да ли купац има наградни купон (1 ако има, 0 ако нема). На стандардни излаз исписати цену коју купац треба да плати након примене свих попушта.

Напомена: Уколико цена након снижења није цео број, резултат заокружити навише.

Улаз:	Улаз:	Улаз:	Улаз:
12 1200 0	6 900 0	9 1000 1	2 800 1
Излаз:	Излаз:	Излаз:	Излаз:
840	810	100	300

Решење

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main() {
7     int br, c, k;
8     cin >> br >> c >> k;
9     if(br<=0 || c <= 0) {
10         cout << "-1" << endl;
11         return -1;
12     }
13
14     int kupon;
15     if(k == 0) kupon = 0;
16     else if(k == 1) {
17         if(br%2==0) kupon = 500;
18         else kupon = 700;
19     }
20     else {
21         cout << "-1" << endl;
22         return -1;
23     }
24
25     float offset;
26     if(br>=10) offset = 0.7;
27     else if(br>=7) offset = 0.8;
28     else if(br>=3) offset = 0.9;
29     else offset = 1.0;
30
31     int finalna_cena = ((int)ceil(c*offset)) - kupon;
32     cout << (finalna_cena>0 ? finalna_cena : 0) << endl;
33
34 }
```

V

5.1

Написати програм који израчунава збир цифара датог целог броја.

Са стандардног улаза се учитава један цео број.

На стандардни излаз исписати збир цифара учитаног броја.

Улаз: 1234 Излаз: 10	Улаз: -12121 Излаз: 7	Улаз: 900000 Излаз: 9	Улаз: -1 Излаз: 1
-----------------------------------	------------------------------------	------------------------------------	--------------------------------

Решење

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     if(n<0) n*=-1;
9
10    int sum = 0;
11    while(n!=0) {
12        sum += n%10;
13        n/=10;
14    }
15    cout << sum << endl;
16 }
```

5.2

Написати програм који у датом тексту проналази најдужу реч, као и број појављивања те речи. Са стандардног улаза се читавају речи (ниске карактера без белина) све до краја улаза. На стандардни излаз исписати најдужу реч и број појављивања те речи. Уколико у тексту постоји више различитих речи максималне дужине, исписати ону која се прва појављује.

Улаз:
ovo je jednostavan tekst

Излаз:
jednostavan 1

Улаз:
pas zec noj

Излаз:
pas 1

Улаз:
abc abc abc

Излаз:
abc 3

Улаз:
algoritam resava problem i algoritam
mora biti ispravan

Излаз:
algoritam 2

Решење

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     string dug = "";
7     int ponavljanje = 0;
8
9     string trenutni;
10    while(cin >> trenutni) {
11        if(trenutni.size() > dug.size()) {
12            dug = trenutni;
13            ponavljanje = 1;
14        }
15        else if(trenutni == dug) {
16            ponavljanje++;
17        }
18    }
19
20    cout << dug << " " << ponavljanje << endl;
21 }
```

VI

6.1

Са стандардног улаза учитавају се број који представља површину једне државе и број n . Затим се учитавају подаци о n држава и то: назив државе (ниска без белина), њена површина (у километрима квадратним) и број становника (два цела броја).

На стандардни излаз исписати просечан број становника у државама које имају површину већу од тражене и, након тога, у новом реду, исписати просечан број становника у државама које имају површину мању од тражене. Резултате заокружити на две децимале. У случају да не постоји ниједна држава са већом (или мањом) површином, исписати 0. У случају грешке на стандардни излаз исписати -1.

Улаз:

644329

4

Egipat 1010408 109450000

Rumunija 238397 19036031

Urugvaj 176215 3499451

Kanada 9984670 41651653

Излаз:

75550826.50

11267741.00

Улаз:

25713

3

Nemacka 357022 83491249

Svedska 450295 10588230

Andora 467 88941

Излаз:

47039739.50

88941.00

Улаз:

25713

1

Andora 467 88941

Излаз:

0.00

88941.00

Улаз:

644329

4

Egipat 1010408 109450000

Rumunija 238397 19036031

Urugvaj 176215 -3499451

Kanada 9984670 41651653

Излаз:

-1

Решење

```
1 #include <iostream>
2 #include <iomanip>
3
4 using namespace std;
5
6 struct Avg {
7     int sum = 0;
8     int count = 0;
9
10    double compute() {
11        return count==0 ? 0.0 : (sum/((double)count));
12    }
13 };
14
15 int main() {
16     int p, n;
17     cin >> p >> n;
18     if(p<0||n<0) {
19         cout << "-1" << endl;
20         return -1;
21     }
22
23     Avg bigger;
24     Avg smaller;
25
26     for(int i = 0; i < n; i++) {
27         string ime_drzave;
28         int p1, s1;
29         cin >> ime_drzave >> p1 >> s1;
30         if(p1<0 || s1<0) {
31             cout << "-1" << endl;
32             return -1;
33         }
34
35         if(p1<p) {
36             smaller.sum += s1;
37             smaller.count++;
38         }
39         else if(p1>p) {
40             bigger.sum += s1;
41             bigger.count++;
42         }
43     }
44
45     cout << fixed << setprecision(2) << bigger.compute()
46         << endl << smaller.compute() << endl;
47 }
```

6.2

Написати програм који проверава да ли на плејлисти постоји песма датог изводача која траје дуже од задатог времена. Са стандардног улаза уносе се назив изводача (ниска без белина) и време трајања песме у секундама. Након тога уноси се број песама на плејлисти - n и подаци о n песама са листе. За сваку песму уноси се име изводача и трајање песме у облику *минут:секунд*.

На стандардни излаз исписати **da** уколико постоји песма која задовољава дате критеријуме. У супротном исписати **ne**.

Улаз:

Queen 120
4
Pantera 5:47
Boston 4:46
Queen 1:55
Queen 6:00

Излаз:

da

Улаз:

Queen 365
4
Pantera 5:47
Boston 4:46
Queen 1:55
Queen 6:00

Излаз:

ne

Улаз:

Stromae 189
4
Stromae 3:10
Stromae 3:06
Queen 1:55
Queen 6:00

Излаз:

da

Улаз:

Stromae 190
2
Pantera 5:47
Boston 4:46

Излаз:

ne

Решење

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     string izvodjac; int trajanje; int n;
7     cin >> izvodjac >> trajanje >> n;
8
9     for(int i = 0; i < n; i++) {
10         string izv; int t_m, t_s; char term;
11
12         cin >> izv >> t_m >> term >> t_s;
13         if(izv==izvodjac && (t_m*60+t_s)>trajanje) {
14             cout << "da" << endl;
15             return 0;
16         }
17     }
18
19     cout << "ne" << endl;
20     return 0;
21 }
```

VII

7.1

Деда Мраз је веома заузет за Нову годину. Да би успео свој деци да донесе поклоне, одлучио је да увежба спуштање низ димњак. У те сврхе, упослио је своје вилењаке да мере колико му времена треба сваки пут када се спусти низ димњак. На крају, да би што боље испланирао своје време, он жели да израчуна просечно време спуштања низ димњак (изражено у секундама).

Са стандардног улаза се уноси број спуштања низ димњак n ($1 \leq n \leq 100$), а затим и n времена које су вилењаци измерили у формату *минути:секунде*. На стандардни излаз исписати просечно време спуштања низ димњак изражено у секундама и заокружено на 2 децимале. У случају неисправног улаза на стандардни излаз исписати -1 .

Улаз:	Улаз:	Улаз:	Улаз:
4	5	5	5
12:22	0:24	0:24	0:24
4:11	1:52	1:52	1:52
9:53	4:22	-4:22	-4:22
12:52	0:52	0:52	0:52
	3:06	3:06	3:06
Излаз:	Излаз:	Излаз:	Излаз:
589.50	127.20	683.00	-1

Решење

```
1 #include <iostream>
2 #include <iomanip>
3
4 using namespace std;
5
6 int main() {
7     int n;
8     cin >> n;
9     if(n<1||n>100) {
10         cout << "-1" << endl;
11         return -1;
12     }
13
14     int sum = 0;
15     for(int i = 0; i < n; i++) {
16         int m, s;
17         char term;
18         cin >> m >> term >> s;
19         if(m < 0 || s < 0 || s >= 60) {
20             cout << "-1" << endl;
21             return -1;
22         }
23         sum += m*60 + s;
24     }
25
26     cout << fixed << setprecision(2) << (sum / ((double) n)) << endl;
27 }
```

7.2

У једној трговини је на снази новогодишња акција. Акција предвиђа да, уколико купац купи барем 3 артикла, трећи најскупљи артикал по цени добија за 1 динар.

Са стандардног улаза се учитава број артикала n ($1 \leq n \leq 100$), а затим и n артикала (позитивни цели бројеви). На стандардни излаз исписати укупну цену артикала након примене новогодишње акције. У случају погрешног улаза, на стандардни излаз исписати -1 .

Улаз:

5

250 500 100 300 400

Излаз:

1251

Улаз:

5

1000 600 800 500 800

Излаз:

2901

Улаз:

2

200 300

Излаз:

500

Улаз:

4

500 -200 100 200

Излаз:

-1

Решење

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4
5 using namespace std;
6
7 bool veci(int a, int b) {
8     return a>b;
9 }
10
11 int main() {
12     int n;
13     cin >> n;
14     if(n<1 || n>100) {
15         cout << "-1" << endl;
16         return -1;
17     }
18
19     vector<int> artikli(n);
20     for(int i = 0; i < n; i++) {
21         int x;
22         cin >> x;
23         if(x<=0) {
24             cout << "-1" << endl;
25             return -1;
26         }
27         artikli[i] = x;
28     }
29
30     sort(begin(artikli), end(artikli), veci);
31     if(n>=3) artikli[2] = 1;
32
33     int sum = 0;
34     for(int i = 0; i < n; i++) {
35         sum += artikli[i];
36     }
37
38     cout << sum << endl;
39
40 }
```

VIII

8.1

Написати програм који за две задате речи проверава да ли друга садржи прву као подреч. Са стандардног улаза задају се две ниске без белина, свака у новом реду. На стандардни излаз исписати да уколико прва ниска представља подреч друге, не у супротном.

Улаз:

abc
abcdef

Излаз:

da

Улаз:

abc
aabbcc

Излаз:

ne

Улаз:

abc
abc

Излаз:

da

Улаз:

abc
aabbabcd

Излаз:

da

Решење

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     string podrec, rec;
7     cin >> podrec >> rec;
8
9     int podrec_n = podrec.size(); int rec_n = rec.size();
10    if(rec_n >= podrec_n) {
11        int j = 0;
12        for(int i = 0; i < rec_n; i++) {
13            if(rec[i] != podrec[j]) j = 0;
14            else {
15                j++;
16                if(j==podrec_n) {
17                    cout << "da" << endl;
18                    return 0;
19                }
20            }
21        }
22    }
23    cout << "ne" << endl;
24    return 0;
25 }
```

8.2

Написати програм који одређује колико студената је на сваком домаћем задатку тачно урадило најмање један задатак.

Са стандардног улаза задају се два цела броја, број студената n и број домаћих задатака k , затим следи n редова, по један за сваког студента. Сваки од тих редова садржи тачно k целих бројева, где сваки број представља колико је задатака тај студент тачно урадио на одговарајућем домаћем задатку. На стандардни излаз исписати број студената који су на сваком домаћем задатку урадили бар један задатак тачно.

Улаз:

```
4
3
1 0 2
1 1 1
2 2 2
1 1 0
```

Излаз:

2

Улаз:

```
2
3
0 0 0
1 2 0
```

Излаз:

0

Улаз:

```
2
3
1 1 1
1 2 3
```

Излаз:

2

Улаз:

```
1
5
1 2 1 2 2
```

Излаз:

1

Решење

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int n, k;
7     cin >> n >> k;
8     if(n<=0 || k <=0) {
9         cout << "0" << endl;
10        return -1;
11    }
12
13    int br = 0;
14    for(int i = 0; i < n; i++) {
15        bool valja = true;
16        for(int j = 0; j < k; j++) {
17            int x;
18            cin >> x;
19            if(x<=0) valja = false;
20        }
21        if(valja) br++;
22    }
23
24    cout << br << endl;
25 }
```

IX

9.1

Написати програм који циклично ротира елементе низа за k позиција удесно.

Са стандардног улаза учитава се цео број n , а затим и n елемената низа. Након тога, у новом реду се уноси цео број k .

У случају исправних аргумената на стандардни излаз исписати резултујући низ. Иначе исписати -1 .

Улаз: 5 1 2 3 4 5 2 Излаз: 4 5 1 2 3	Улаз: 5 1 2 3 4 5 5 Излаз: 1 2 3 4 5	Улаз: 5 1 2 3 4 5 7 Излаз: 4 5 1 2 3	Улаз: -5 1 2 3 4 5 5 Излаз: -1
---	---	---	---

Решење

```
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 int main() {
7     int n;
8     cin >> n;
9     if(n<=0) {
10         cout << "-1" << endl;
11         return -1;
12     }
13
14     vector<int> niz(n);
15     for(int i = 0; i < n; i++) {
16         cin >> niz[i];
17     }
18
19     int k;
20     cin >> k;
21
22     k%=n;
23     if(k<0) k += n;
24
25     if(k!=0) k = n - k;
26
27     cout << niz[k];
28     for(int i = 1; i < n; i++) {
29         cout << " " << niz[(i+k)%n];
30     }
31     cout << endl;
32 }
```

9.2

Написати програм који испитује колико има парних елемената пре максималног елемента низа, а колико непарних после минималног.

Са стандардног улаза уносе се елементи низа до краја улаза. На стандардни излаз исписати број парних елемената низа који претходе максималном елементу низа, а затим и број непарних елемената низа који се налазе иза минималног елемента. У случају да има више минималних или максималних елемената, у обзир узети последњи.

Улаз:
4 7 2 8 -2 5

Излаз:
2
1

Улаз:
1 1 5 6 4 6

Излаз:
2
1

Улаз:
5 4 3 2 1

Излаз:
0
0

Улаз:
4 4 4 4 4

Излаз:
4
0

Улаз:
1 1 1 1 1

Излаз:
0
0

Решење

```
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 int main() {
7
8     vector<int> niz;
9     {
10         int x;
11         while(cin >> x) niz.push_back(x);
12     }
13
14     int n = niz.size();
15     if(n<=1) {
16         cout << "0" << endl << "0" << endl;
17         return 0;
18     }
19
20     int uk_parnih = 0;
21     int parnih_pre = 0;
22     int max = niz[0];
23
24     int neparnih_posle = 0;
25     int min = niz[0];
26
27     for(int i = 0; i < n; i++) {
28         int x = niz[i];
29
30         if(x%2==1) neparnih_posle++;
31
32         if(max<=x) {
33             max = x;
34             parnih_pre = uk_parnih;
35         }
36
37         if(min>=x) {
38             min = x;
39             neparnih_posle = 0;
40         }
41
42         if(x%2==0) uk_parnih++;
43     }
44
45     cout << parnih_pre << endl << neparnih_posle << endl;
46 }
```

X

10.1

На Олимпијади су учествовали такмичари из многих земаља. Дат је списак земаља учесница, при чему је за сваку земљу познат број освојених медаља и број такмичара. Написати програм који сортира земље по следећим критеријумима (од најважнијег до најмање важног):

- број медаља (нерастуће)
- број такмичара (неоппадајућа)
- име земље (лексикографски растуће)

Са стандардног улаза се учитава број земаља n , а затим у наредних n редова подаци о земљама. За сваку земљу се учитава назив (ниска без белина), број медаља и број такмичара. На стандардни излаз исписати називе земаља након сортирања. У случају неисправног улаза, на стандардни излаз исписати -1 .

Улаз:

4

Kina 120 520

SAD 150 430

Austrija 20 120

Italija 40 200

Излаз:

SAD

Kina

Italija

Australija

Улаз:

5

Kina 150 520

SAD 150 430

Austrija 20 120

Italija 40 200

Francuska 40 220

Излаз:

SAD

Kina

Italija

Francuska

Austrija

Улаз:

5

Kina 150 520

SAD 150 520

Italija 45 200

Austrija 50 120

Francuska 40 220

Излаз:

Kina

SAD

Austrija

Italija

Francuska

```
Улаз:
5
Kina 150 520
SAD 150 -520
Italija 45 200
Austrija 50 120
Francuska 40 220

Излаз:
-1
```

Решење

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4
5 using namespace std;
6
7 struct Kandidat {
8     string name;
9     int br_medalja;
10    int br_takmicara;
11 };
12
13 bool uporedi(Kandidat &a, Kandidat &b) {
14     if(a.br_medalja>b.br_medalja) return true;
15     else if(a.br_medalja<b.br_medalja) return false;
16     else if(a.br_takmicara < b.br_takmicara) return true;
17     else if(a.br_takmicara > b.br_takmicara) return false;
18     else return a.name < b.name;
19 }
20
21 int main() {
22     int n;
23     cin >> n;
24     if(n<0) {
25         cout << "-1" << endl;
26         return -1;
27     }
28
29     vector<Kandidat> niz(n);
30     for(int i = 0; i < n; i++) {
31         Kandidat curr;
32         cin >> curr.name >> curr.br_medalja >> curr.br_takmicara;
33         if(curr.br_medalja < 0 || curr.br_takmicara < 0) {
34             cout << "-1" << endl;
35             return -1;
36         }
37         niz[i] = curr;
38     }
39
40     sort(begin(niz), end(niz), uporedi);
41
42     for(int i = 0; i < n; i++) {
43         cout << niz[i].name << endl;
44     }
45 }
```

10.2

Написати програм који проверава да ли је дати квадрат целих бројева магичан. Квадрат је магичан ако су му једнаки зборови свих врста, свих колона, као и главне и споредне дијагонале.

Са стандардног улаза се учитава димензија квадрата n ($2 \leq n \leq 10$), а затим и квадрат целих бројева димензија $n \times n$. На стандардни излаз исписати **da** уколико је квадрат магичан, а иначе **ne**. Подразумевати да је улаз исправно задат.

Улаз:

3
2 7 6
9 5 1
4 3 8

Излаз:

da

Улаз:

3
1 2 3
3 1 2
2 3 1

Излаз:

ne

Улаз:

4
8 -6 -5 5
-3 3 2 0
1 -1 -2 4
-4 6 7 -7

Излаз:

da

Улаз:

4
7 -6 -5 5
-3 3 2 0
1 -1 -2 4
-4 6 7 -7

Излаз:

ne

Решење

```
1 #include <iostream>
2 #include <vector>
3
4 using namespace std;
5
6 int main() {
7     int n;
8     cin >> n;
9     if(n<2 || n>10) {
10         cout << "-1" << endl;
11         return -1;
12     }
13
14     vector<vector<int>> mat(n,vector<int>(n));
15     for(int i = 0; i < n; i++) {
16         for(int j = 0; j < n; j++) {
17             cin >> mat[i][j];
18         }
19     }
20
21     int zbir = 0;
22     for(int i = 0; i < n; i++) zbir += mat[0][i];
23
24     for(int i = 1; i < n; i++) {
25         int z = 0;
26         for(int j = 0; j < n; j++) z += mat[i][j];
27         if(z!=zbir) {
28             cout << "ne" << endl;
29             return 0;
30         }
31     }
32
33     for(int i = 0; i < n; i++) {
34         int z = 0;
35         for(int j = 0; j < n; j++) z += mat[j][i];
36         if(z!=zbir) {
37             cout << "ne" << endl;
38             return 0;
39         }
40     }
41
42     int d1 = 0;    int d2 = 0;
43
44     for(int i = 0; i < n; i++) {
45         d1 += mat[i][i];
46         d2 += mat[i][n-1-i];
47     }
48
49     if(d1!=zbir || d2!=zbir) {
50         cout << "ne" << endl;
51         return 0;
52     }
53
54     cout << "da" << endl;
55     return 0;
56 }
```