

Општинско такмичење из програмирања за ученике основних школа

4. март 2017.

5. разред

1. Ада жели да за Нову годину украси правоугаони прозор ширине s и висине v центиметара, тако што ће око њега обмотати украсну траку. Написати програм UKRAS којим се одређује колика је дужина украсне траке коју Ада треба да купи. У првој линији стандардног улаза се налази целобројна вредност s , а у следећој линији целобројна вредност v ($80 < s \leq 200$, $100 < v \leq 150$) које редом представљају ширину и висину прозора изражену у центиметрима. У првој линији стандардног излаза исписати један цео број који представља дужину украсне траке у центиметрима.

ПРИМЕР

УЛАЗ	ИЗЛАЗ
100	450
125	

2. Студент Марко је добио плаћену програмерску праксу у компанији која прави рачунарске игре. која ће трајати тачно n месеци. Првог месеца ће му плата бити a динара, а ако буде вредно радио, сваког наредног месеца ће добијати повишицу од d динара. Знајући да је Марко сјајан програмер и да ће веома вредно радити, напиши програм PRAKSA који израчунава колика ће му бити плата у последњем месецу праксе. Уносе се три природна броја, сваки у посебној линији стандардног улаза. Бројеви редом представљају број месеци колико пракса траје n ($1 \leq n \leq 12$), колико је Маркова плата првог месеца праксе a ($3000 \leq a \leq 40000$), и колика је повишица сваког месеца d ($500 \leq d \leq 5000$). На стандардном излазу приказати колику плату Марко добије последњег месеца праксе.

ПРИМЕР

УЛАЗ	ИЗЛАЗ
10	14000
5000	
1000	

3. За дате природне бројеве a и b написати програм RAZLOMAK којим се дати неправи разломак $\frac{a}{b}$ преводи у мешовит број $n\frac{c}{b}$ такав да важи да је $c < b$. У првој линији стандардног улаза налази се природан број a ($a \leq 1000$) који представља бројилац неправог разломка, а у другој линији природан број b различит од нуле који представља именилац разломка ($b \leq a$). Прва и једина линија стандардног излаза треба да садржи мешовити запис разломка, прецизније природан број, бројилац и именилац мешовитог броја међусобно одвојени са по једном празнином (бланко знаком).

ПРИМЕР

УЛАЗ	ИЗЛАЗ
23	2 7 8
8	

4. Ако је дат угао који сатна казаљка заклапа са вертикалном полуправом која спаја центар и врх сата (број 12), написати програм KAZALJKE који ће одредити време у сатима и минутима. Са стандардног улаза читава се један природан број a ($a < 360$) који представља угао у степенима. На стандардни излаз исписати у једном реду два цела броја h ($0 \leq h < 12$) и m ($0 \leq m < 60$) раздвојена двотачком, који представљају број сати и минута који одговарају положају сатне казаљке.

ПРИМЕР

УЛАЗ	ИЗЛАЗ
124	4:8

Израда задатака траје 120 минута

1. задатак – 25 поена
2. задатак – 25 поена
3. задатак – 25 поена
4. задатак – 25 поена

**Општинско такмичење из програмирања за ученике основних школа
4. март 2017.**

5. разред – тест примери

Пет исправних тест примера у задатку ученику доносе 25 поена за тај задатак, четири исправна тест примера ученику доносе 15 поена за тај задатак, три исправна тест примера ученику доносе 5 поена за тај задатак, док мање од три исправна тест примера ученику не доносе поене.

1. UKRAS

УЛАЗ	ИЗЛАЗ
103 114	434
80 100	360
200 150	700
193 122	630
165 118	566

2. PAKSA

УЛАЗ	ИЗЛАЗ
2 10000 1500	11500
8 28345 1235	36990
12 34000 1300	48300
7 12345 876	17601
11 33333 1111	44443

3. RAZLOMAK

УЛАЗ	ИЗЛАЗ
17 3	5 2 3
123 4	30 3 4
456 7	65 1 7
789 8	98 5 8
456 345	1 111 345

4. KAZALJKE

УЛАЗ	ИЗЛАЗ
89	2:58
91	3:2
179	5:58
195	6:30
273	9:6

1. UKRAS

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

int main()
{
    int s, v;
    cin >> s >> v;
    int O = 2 * s + 2 * v;
    cout << O << endl;
    return 0;
}
```

2. PAKSA

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

int main() {
    int n, a, d;
    cin >> n >> a >> d;
    int x = a + (n - 1) * d;
    cout << x << endl;
    return 0;
}
```

3. RAZLOMAK

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

int main() {
    int a, b;
    cin >> a >> b;
    int n = a / b;
    int c = a % b;
    cout << n << " " << c << " " << b << endl;
    return 0;
}
```

4. KAZALJKE

Прво решење:

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

// za dati broj sati i minuta odredjuje broj minuta proteklih od prethodnih 00:00 sati
int hm_to_M(int h, int m) {
    return 60*h + m;
}

// za dati broj minuta proteklih od prethodnih 00:00 sati odredjuje broj sati i minuta
void M_to_hm(int M, int& h, int& m) {
    m = M % 60;
    h = M / 60;
}

int main() {
    int alfa;
    cin >> alfa;
    // M - broj minuta proteklih od prethodnih 00:00 sati
    // vazi proporcija da je M : alfa = hm_to_M(12, 00) : 360 stepeni
    int M = alfa * hm_to_M(12, 00) / 360;
    // broj sati i minuta koji odgovaraju tom broju minuta
    int h, m;
    M_to_hm(M, h, m);
    // ispis rezultata
    cout << h << ":" << m << endl;
    return 0;
}
```

Друго решење:

За 12 сати сатна кажалка пређе угао од 360 степени, па за један сат сатна пређе угао од 30 степени. Зато за један минут пређе угао од пола степена. Зато је број временских минута двоструко већи од датог броја угаоних степени.

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

// za dati broj minuta proteklih od prethodnih 00:00 sati odredjuje broj sati i minuta
void M_to_hm(int M, int& h, int& m) {
    m = M % 60;
    h = M / 60;
}

int main() {
    int alfa;
    cin >> alfa;
    // pošto satna kazaljka za jedan minut pređe pola ugaona stepena,
    // broj proteklih minuta jednak je dvostrukom broju ugaonih stepeni
    int M = alfa * 2;
    // broj sati i minuta koji odgovaraju tom broju minuta
    int h, m;
    M_to_hm(M, h, m);
    // ispis rezultata
    cout << h << ":" << m << endl;
    return 0;
}
```

Општинско такмичење из програмирања за ученике основних школа

4. март 2017.

6. разред

1. Ада жели да за Нову годину украси правоугаони прозор ширине s и висине v центиметара, тако што ће око њега обмотати украсну траку. Написати програм UKRAS којим се одређује колика је дужина украсне траке коју Ада треба да купи. У првој линији стандардног улаза се налази целобројна вредност s , а у следећој линији целобројна вредност v ($80 < s \leq 200$, $100 < v \leq 150$) које редом представљају ширину и висину прозора изражену у центиметрима. У првој линији стандардног излаза исписати један цео број који представља дужину украсне траке у центиметрима.

ПРИМЕР

УЛАЗ	ИЗЛАЗ
100	450
125	

2. Сваког јутра Перица за доручак купује три кроасана и чоколадно млеко. Чоколадно млеко је s динара скупље од једног кроасана (све цене су изражене као природни бројеви). Напиши програм PEKARA којим се проверава да ли унети број d може бити укупна цена Перициног доручка. Са стандардног улаза уносе се два природна броја s ($s \leq 50$) и d ($d \leq 500$). На стандардни излаз исписати реч `da` ако је Перица могао платити доручак d динара, тј. `ne` ако није.

ПРИМЕР

УЛАЗ	ИЗЛАЗ
3	ne
85	
3	da
83	

3. Јован, Лука и Пера су три рођена брата. Напиши програм GODINE који их уређује од најстаријег до најмлађег. Са стандардног улаза уносе се три цела броја j , l и p из интервала $[0, 100]$, сваки у посебном реду. На стандардни излаз исписују се бројеви уређени од највећег до најмањег.

ПРИМЕР

УЛАЗ	ИЗЛАЗ
7	9
5	7
9	5

4. Написати програм UGAO који ће одредити угао у степенима и минутима између сатне и минутне казаљке ако је задато време у сатима и минутима. Угао је увек позитиван и мери се у смеру кретања казаљки на сату (зато може да буде и већи од 180 степени). Учитавају се два броја, сваки у посебној линији. У првој линији је број сати (између 0 и 12), а у другој линији је број минута (између 0 и 59). На стандардни излаз исписати степен угла 0 до 359, и минут угла од 0 до 59, раздвојене двотачком тј. карактером `:`.

ПРИМЕР

УЛАЗ	ИЗЛАЗ
12	165:0
30	

Израда задатака траје 120 минута

1. задатак – 25 поена
2. задатак – 25 поена
3. задатак – 25 поена
4. задатак – 25 поена

Општинско такмичење из програмирања за ученике основних школа
4. март 2017.
6. разред – тест примери

Пет исправних тест примера у задатку ученику доносе 25 поена за тај задатак, четири исправна тест примера ученику доносе 15 поена за тај задатак, три исправна тест примера ученику доносе 5 поена за тај задатак, док мање од три исправна тест примера ученику не доносе поене.

1. UKRAS

УЛАЗ	ИЗЛАЗ
103 114	434
80 100	360
200 150	700
193 122	630
165 118	566

2. PEKARA

УЛАЗ	ИЗЛАЗ
15 105	ne
15 115	da
10 94	da
10 105	ne
11 99	da

3. KUPOVINA

УЛАЗ	ИЗЛАЗ
10 20 30	30 20 10
55 25 35	55 35 25
30 20 10	30 20 10
22 42 32	42 32 22
30 35 25	35 30 25

4. UGAO

УЛАЗ	ИЗЛАЗ
1 30	135:0
4 47	138:30
10 18	201:0
0 38	209:0
11 29	170:30

1. UKRAS

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
{
    int s, v;
    cin >> s >> v;
    int O = 2 * s + 2 * v;
    cout << O << endl;
    return 0;
}
```

2. PEKARA

```
#include <iostream>
using namespace std;
int main() {
    int s, d;
    cin >> s >> d;
    cout << ((d - s) % 4 == 0 ? "da" : "ne") << endl;
    return 0;
}
```

3. GODINE

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

// uredjivanje selekcijom
void uredi(int& a, int& b, int& c) {
    if (a < b) swap(a, b);
    if (a < c) swap(a, c);
    if (b < c) swap(b, c);
}

int main() {
    // godine tri brata
    int g1, g2, g3;
    cin >> g1 >> g2 >> g3;
    uredi(g1, g2, g3);
    cout << g1 << endl << g2 << endl << g3 << endl;
    return 0;
}
```

4. UGAO

```
#include <iostream>
#include <cassert>
#include <cmath>
using namespace std;
int main() {
    // učitavamo vreme
    int sat, minut;
    cin >> sat >> minut;
    sat %= 12; // sat svodimo na interval [0, 12)
    // ugao u minutima koji satna kazaljka zauzima sa položajem 12h
    int ugaoniMinutiSatne = sat * 30 * 60 + minut * 30;
    // ugao u minutima koji minutna kazaljka zauzima sa položajem 12h
    int ugaoniMinutiMinutne = minut * 360;
    // ugao između satne i minutne kazaljke u minutima
    int ugaonaRazlikaUMinutima =
        abs(ugaoniMinutiSatne - ugaoniMinutiMinutne);
    // ugao između kazaljke u stepenima i minutima
    int stepenUgaoneRazlike = ugaonaRazlikaUMinutima / 60;
    int minutUgaoneRazlike = ugaonaRazlikaUMinutima % 60;
    // ispis rezultata
    cout << stepenUgaoneRazlike << ":" << minutUgaoneRazlike << endl;
}
```

Општинско такмичење из програмирања за ученике основних школа

4. март 2017.

7. разред

1. Сваког јутра Перица за доручак купује три кроасана и чоколадно млеко. Чоколадно млеко је s динара скупље од једног кроасана (све цене су изражене као природни бројеви). Напиши програм **PEKARA** којим се проверава да ли унети број d може бити укупна цена Перициног доручка. Са стандардног улаза уносе се два природна броја s ($s \leq 50$) и d ($d \leq 500$). На стандардни излаз исписати реч да ако је Перица могао платити доручак d динара, тј. не ако није.

ПРИМЕР

УЛАЗ

3

85

3

83

ИЗЛАЗ

ne

da

2. Милица купује патике. Допала су јој се три пара. Напиши програм **KUPOVINA** који исписује цене та три пара патика од најскупљих до најјефтинијих. Са стандардног улаза уносе се три цела броја из интервала $[1000, 10000]$, сваки у посебном реду.

ПРИМЕР

УЛАЗ

2017

2000

2016

ИЗЛАЗ

2017

2016

2000

3. Дата су два угла разностраничног троугла у степенима, минутима и секундама. Написати програм **TROUGAO** који одређује који од углова троугла је највећи, први по редоследу уноса, други по редоследу уноса или трећи, чија се вредност се одређује у програму. У свакој од шест линија стандардног улаза налази се по један цео број. Бројеви редом представљају степене, минуте (мање од 60) и секунде (мање од 60), прво једног, па другог угла троугла. Збир дата два угла је мањи од 180 степени. У једној линији стандардног излаза исписати број 1, 2 или 3, а у другој величину тог највећег угла у степенима, минутима и секундама (три броја одвојена размаком).

ПРИМЕР

УЛАЗ

75

30

14

23

15

45

ИЗЛАЗ

3

81 14 1

4. Два другара возе бицикл по кружној стази дужине s km, један брзином од v_1 km/h, други брзином од v_2 km/h. Написати програм **TRKA** који ће израчунати колико ће времена требати бржем тркачу да за цео круг прстигне споријег (ако су истовремено кренули са старта)? У прве три линије стандардног улаза се налазе редом по један податак, реалан број заокружен на две децимале: s - дужина кружне стазе ($1 \leq s \leq 10$), v_1 - брзина првог тркача ($1 \leq v_1 \leq 45$), v_2 - брзина другог тркача ($1 \leq v_2 \leq 45$). Исписати на стандардни излаз у сваком реду по једну целобројну вредност броја сати, минута и секунди, који представљају време потребно да бржи тркач за цео круг сустигне споријег. Секунде приказати заокружене на најближу целобројну вредност.

ПРИМЕР

УЛАЗ

10.00

5.00

7.00

ИЗЛАЗ

5

0

0

Израда задатака траје 120 минута

1. задатак – 25 поена

2. задатак – 25 поена

3. задатак – 25 поена

4. задатак – 25 поена

Општинско такмичење из програмирања за ученике основних школа
4. март 2017.

7. разред – тест примери

Пет исправних тест примера у задатку ученику доносе 25 поена за тај задатак, четири исправна тест примера ученику доносе 15 поена за тај задатак, три исправна тест примера ученику доносе 5 поена за тај задатак, док мање од три исправна тест примера ученику не доносе поене.

1. PEKARA

УЛАЗ	ИЗЛАЗ
15 105	ne
15 115	da
10 94	da
10 105	ne
11 99	da

2. KUPOVINA

УЛАЗ	ИЗЛАЗ
1000	3000
2000	2000
3000	1000
5000	5000
2000	3000
3000	2000
3000	3000
2000	2000
1000	1000
2000	9000
9000	3000
3000	2000
3000	5000
5000	3000
1000	1000

Општинско такмичење из програмирања за ученике основних школа
4. март 2017.
7. разред

1. PEKARA

```
#include <iostream>
using namespace std;
int main() {
    int s, d;
    cin >> s >> d;
    cout << ((d - s) % 4 == 0 ? "da" : "ne") << endl;
    return 0;
}
```

2. KUPOVINA

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

// uredjuje tri broja opadajuće
void uredi(int& a, int& b, int& c) {
    if (a > b) swap(a, b);
    if (a > c) swap(a, c);
    if (b > c) swap(b, c);
}

int main() {
    // cene tri para patika
    int p1, p2, p3;
    cin >> p1 >> p2 >> p3;
    uredi(p1, p2, p3);
    cout << p1 << endl << p2 << endl << p3 << endl;
    return 0;
}
```

3. TROUGAO

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

// prevodi ugao od d stepeni, m minuta i s sekundi u sekunde
int USEkunde(int d, int m, int s) {
    return 60*60*d + 60*m + s;
}

int main(){
    int d1, m1, s1, d2, m2, s2;
    cin >> d1 >> m1 >> s1;
    cin >> d2 >> m2 >> s2;
    // izracunavamo velicinu sva tri ugla samo u sekundama
    int sec1 = USEkunde(d1, m1, s1);
    int sec2 = USEkunde(d2, m2, s2);
    int sec3 = USEkunde(180, 0, 0) - (sec1 + sec2);
    // odredjujemo poziciju maksimalnog ugla
    int secMax = sec1; int pozMax = 1;
    if (sec2 > secMax) { secMax = sec2; pozMax = 2; }
    if (sec3 > secMax) { secMax = sec3; pozMax = 3; }
    cout << pozMax << endl;

    int s = secMax % 60;
    int m = (secMax / 60) % 60;
    int d = (secMax / (60*60));

    cout << d << " " << m << " " << s << endl;

    return 0;
}
```

4. TRKANJE

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    double s, // duzina kruzne staze u km
           v1, // brzina prvog trkaca u km/h
           v2; // brzina drugog trkaca u km/h
    cin >> s >> v1 >> v2;
    // izracunavamo vreme u satima potrebno da brzi trkac za ceo krug prestigne sporijeg
    double t = s / abs(v1 - v2);
    // prevodimo dobijeno vreme u sate, minute i sekunde
    int sati, minuta, sekundi;
    sekundi = round(t * 3600); // broj sekundi zaokruzen na najblizi ceo broj
    minuta = sekundi / 60;
    sekundi %= 60;
    sati = minuta / 60;
    minuta %= 60;

    // ispisujemo rezultat
    cout << sati << endl << minuta << endl << sekundi << endl;
    return 0;
}
```

Општинско такмичење из програмирања за ученике основних школа
4. март 2017.
8. разред

1. Два другара Пера и Мика су дошла у школу. Ако се за сваког зна сат, минут и секунд када је стигао, напиши програм **DRUGARI** који одређује који од њих је стигао пре. Са стандардног улаза учитава се 6 целих бројева. Прво сат, минут и секунд када је стигао Пера, а затим сат, минут и секунд када је стигао Мика (претпоставља се да бројеви представљају исправно задата времена). На стандардни излаз исписати једну линију текста. Пера ако је пре стигао Пера, Мика ако је пре стигао Мика или *istovremeno* ако су стигли истовремено.

ПРИМЕР

УЛАЗ

14

23

17

15

23

11

ИЗЛАЗ

Pera

2. У једној школи програмирања уведено је правило којим се одређује износ попушта који остварују ученици приликом уписа. Ученици са одличним успехом остварују попуст од 40% укупног износа школарине, са врлодобрим 20% а са добрим 10%. Такође, ученици који су освојили награду на неком од државних такмичења остварују попуст од 30% укупног износа школарине. Уколико неки ученик испуњава два критеријума за попуст примењује се критеријум по коме је попуст већи. На основу пуног износа школарине, просечне оцене ученика и информације о наградама са такмичења написати програм **SKOLA** који ће одредити износ који ученик треба да плати при упису у наредну школску годину. У првој линији стандардног улаза налази се пун износ школарине (реалан број), у другој просечна оцена ученика (реалан број од 2.0 до 5.0) а у трећој 0 уколико ученик нема награду или 1 уколико је има. Износ школарине коју ученик треба да плати (заокружен на најближи цео број) наводи се у једној линије стандардног излаза.

ПРИМЕР

УЛАЗ

4000

4.65

1

ИЗЛАЗ

2400

3. Дата су два угла разностраничног троугла у степенима, минутима и секундама. Написати програм **TROUGAO** који одређује који од углова троугла је највећи, први по редоследу уноса, други по редоследу уноса или трећи, чија се вредност се одређује у програму. У свакој од шест линија стандардног улаза налази се по један цео број. Бројеви редом представљају степене, минуте (мање од 60) и секунде (мање од 60), прво једног, па другог угла троугла. Збир дата два угла је мањи од 180 степени. У једној линији стандардног излаза исписати број 1, 2 или 3, а у другој величину тог највећег угла у степенима, минутима и секундама (три броја одвојена размаком).

ПРИМЕР

УЛАЗ

75

30

14

23

15

45

ИЗЛАЗ

3

81 14 1

4. Два другара возе бицикл по кружној стази дужине s km, један брзином од v_1 km/h, други брзином од v_2 km/h. Написати програм **TRKA** који ће израчунати колико ће времена требати бржем тркачу да за цео круг престигне споријег (ако су истовремено кренули са старта)? У прве три линије стандардног улаза се налазе редом по један податак, реалан број заокружен на две децимале: s - дужина кружне стазе ($1 \leq s \leq 10$), v_1 - брзина првог тркача ($1 \leq v_1 \leq 45$), v_2 - брзина другог тркача ($1 \leq v_2 \leq 45$). Исписати на стандардни излаз у сваком реду по једну целобројну вредност броја сати, минута и секунди, који представљају време потребно да бржи тркач за цео круг сустигне споријег. Секунде приказати заокружене на најближу целобројну вредност.

ПРИМЕР

УЛАЗ

10.00

5.00

7.00

ИЗЛАЗ

5

0

0

Израда задатака траје 120 минута

1. задатак – 25 поена

2. задатак – 25 поена

3. задатак – 25 поена

4. задатак – 25 поена

**Општинско такмичење из програмирања за ученике основних школа
4. март 2017.**

8. разред – тест примери

Пет исправних тест примера у задатку ученику доносе 25 поена за тај задатак, четири исправна тест примера ученику доносе 15 поена за тај задатак, три исправна тест примера ученику доносе 5 поена за тај задатак, док мање од три исправна тест примера ученику не доносе поене.

1. DRUGARI

УЛАЗ	ИЗЛАЗ
12 0 0 12 0 0	istovremeno
8 42 30 6 24 9	Mika
20 45 8 16 56 23	Mika
5 47 20 9 49 30	Pera
15 25 57 20 26 24	Pera

2. SKOLA

УЛАЗ	ИЗЛАЗ
9925.38 2.48 1	6948
5651.71 3.19 1	3956
6499.52 2.17 0	6500
9109.86 4.53 1	5466
6003.01 2.44 0	6003

3. TROUGAO

УЛАЗ	ИЗЛАЗ
29 47 13 144 49 55	2 144 49 55
81 53 47 9 56 49	3 88 9 24
93 50 40 30 59 39	1 93 50 40
26 53 5 17 50 59	3 135 15 56
148 59 15 22 40 44	1 148 59 15

4. TRKANJE

УЛАЗ	ИЗЛАЗ
2.44 44.65 2.74	0 3 30
6.38 24.86 3.52	0 17 56
6.68 19.64 38.06	0 21 46
9.16 19.62 29.84	0 53 47
5.00 35.77 31.60	1 11 57

Општинско такмичење из програмирања за ученике основних школа
4. март 2017.
8. разред

1. DRUGARI

```
#include <iostream>
using namespace std;
int hms_to_S(int h, int m, int s) {
    return h*60*60 + m*60 + s;
}
int main() {
    int hPera, mPera, sPera;
    cin >> hPera >> mPera >> sPera;
    int hMika, mMika, sMika;
    cin >> hMika >> mMika >> sMika;
    // Preracunavamo vremena u broj sekundi od pocetka dana
    int SPera = hms_to_S(hPera, mPera, sPera);
    int SMika = hms_to_S(hMika, mMika, sMika);
    if (SPera < SMika)      cout << "Pera" << endl;
    else if (SMika < SPera)  cout << "Mika" << endl;
    else      cout << "istovremeno" << endl;
    return 0;
}
```

2. SKOLA

```
#include <iostream>
#include <algorithm>
#include <cmath>
using namespace std;
// Za datu skolarinu se odredjuje popust na osnovu uspeha
double PopustUspeh(double skolarina, double prosek)
{
    if (prosek >= 4.5)      return 0.4 * skolarina;
    if (prosek >= 3.5)      return 0.2 * skolarina;
    if (prosek >= 2.5)      return 0.1 * skolarina;
    return 0;
}
// Za datu skolarinu se odredjuje popust na osnovu takmicenja
double PopustTakmicenje(double skolarina, int takmicenje) {
    if (takmicenje != 0)    return 0.3 * skolarina;
    else      return 0.0;
}
int main()
{
    double skolarina;
    cin >> skolarina;
    double prosek;
    cin >> prosek;
    int takmicenje;
    cin >> takmicenje;
    double popust = max(PopustUspeh(skolarina, prosek),
                        PopustTakmicenje(skolarina, takmicenje));
    cout << round(skolarina - popust) << endl;
    return 0;
}
```

3. TROUGAO

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

// prevodi ugao od d stepeni, m minuta i s sekundi u sekunde
int USekunde(int d, int m, int s) {
    return 60*60*d + 60*m + s;
}

int main(){
    int d1, m1, s1, d2, m2, s2;
    cin >> d1 >> m1 >> s1;
    cin >> d2 >> m2 >> s2;
    // izracunavamo velicinu sva tri ugla samo u sekundama
    int sec1 = USekunde(d1, m1, s1);
    int sec2 = USekunde(d2, m2, s2);
    int sec3 = USekunde(180, 0, 0) - (sec1 + sec2);
    // odredjujemo poziciju maksimalnog ugla
    int secMax = sec1; int pozMax = 1;
    if (sec2 > secMax) { secMax = sec2; pozMax = 2; }
    if (sec3 > secMax) { secMax = sec3; pozMax = 3; }
    cout << pozMax << endl;

    int s = secMax % 60;
    int m = (secMax / 60) % 60;
    int d = (secMax / (60*60));

    cout << d << " " << m << " " << s << endl;

    return 0;
}
```

4. TRKANJE

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int main() {
    double s, // duzina krugne staze u km
           v1, // brzina prvog trkaca u km/h
           v2; // brzina drugog trkaca u km/h
    cin >> s >> v1 >> v2;
    // izracunavamo vreme u satima potrebno da brzi trkac za ceo krug prestigne sporijeg
    double t = s / abs(v1 - v2);
    // prevodimo dobijeno vreme u sate, minute i sekunde
    int sati, minuta, sekundi;
    sekundi = round(t * 3600); // broj sekundi zaokruzen na najblizi ceo broj
    minuta = sekundi / 60;
    sekundi %= 60;
    sati = minuta / 60;
    minuta %= 60;
    cout << sati << endl << minuta << endl << sekundi << endl;
    return 0;
}
```

3. TROUGAO

УЛАЗ	ИЗЛАЗ
29 47 13 144 49 55	2 144 49 55
81 53 47 9 56 49	3 88 9 24
93 50 40 30 59 39	1 93 50 40
26 53 5 17 50 59	3 135 15 56
148 59 15 22 40 44	1 148 59 15

4. TRKANJE

УЛАЗ	ИЗЛАЗ
2.44	0
44.65	3
2.74	30
6.38	0
24.86	17
3.52	56
6.68	0
19.64	21
38.06	46
9.16	0
19.62	53
29.84	47
5.00	1
35.77	11
31.60	57

Општинско такмичење из програмирања за ученике основних школа
21. фебруар 2016.
I категорија (5. и 6. разред)

Израда задатака траје 120 минута

1. задатак – 30 поена, 2. задатак – 35 поена, 3. задатак – 35 поена

1. Сунђер Боб има тајни задатак: мора да направи супер плочицу у облику правоугаоника. Познато је колико износи обим плочице, и позната је дужина једне стране плочице. Напишите програм **PRAVOUGAONIK** који ће израчунати дужину друге стране правоугаоника. У једином реду стандардног улаза дата су два природна броја **O**, **A** (међусобно раздвојени једним бланко карактером) који представљају редом обим и дужину једне стране правоугаоника ($4 \leq O \leq 4000$, $1 \leq A \leq 1000$, **O** је паран број). На стандардном излазу исписати број који представља дужину друге стране правоугаоника.

ТЕСТ ПРИМЕР

УЛАЗ	ИЗЛАЗ
12 1	5

РЕШЕЊЕ:

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

```
int main()
{   int o, a, b;
    cin >> o >> a;
    b = (o-2*a)/2;
    cout << b << endl;
    return 0;
}
```

УЛАЗ	ИЗЛАЗ
2364 1000	182
2000 879	121
156 48	30
264 11	121
4000 36	1964
144 1	71

2. Госпођа Пуфна воли да игра игрице, те је на хард диску свог рачунара сместила доста компјутерских игрица. Али, госпођа Пуфна жели да на рачунару сачува и песму свог омиљеног бенда, те мора да израчуна колико има места на хард диску након што је сачувала песму која дугачка **n** минута и **m** секунди. Познато је да преостали слободни простор на хард диску је **k** MB (megabyte), као и да за запис 1 секунде звука је потребно 16 KB (kilobyte). Напишите програм **KONVERZIJA** који исписује (у једном или у више редова) два броја: колико је MB и колико KB простора преостало на хард диску. Сетите се да 1 MB = 1024 KB (1 megabyte има 1024 kilobyte). У једином реду стандардног улаза дати су цели бројеви **n**, **m**, **k** ($1 \leq n, m, k \leq 1000000$) међусобно раздвојени једним бланко карактером. Сматрати да ће крајњи резултат увек чинити ненегативни бројеви (бројеви који нису мањи од 0).

ТЕСТ ПРИМЕР

УЛАЗ	ИЗЛАЗ
5 28 32	26 896

Објашњење тест примера: n=5 минута m=28 секунди k=32 MB

5 минута 28 секунди= 328 секунде (и за 328 секунде потребно је 5248 KB меморије)

На хард диску је пре снимања песме било k=32MB=32768 KB

Након што Пуфна сними песмицу преостаће 32768KB-5248KB=27520 KB

Када претворимо тај број у MB и KB биће: 26 MB и 896 KB

РЕШЕЊЕ:

Да би се решио задатак, потребно је искористити однос **јединица мере** $1 \text{ MB} = 1024 \text{ KB}$.
Ако *vreme* је време трајања песме, и ако *memorija* је капацитет аудио датотеке (изражен у KB), онда важи да $vreme = 60 * n + m$; $memorija = 16 * vreme$;

```
#include <iostream>
using namespace std;
int main()
{ int n,m,k;
cin >> n >> m >> k;
long long vreme,memorija, rezultat;
long long rez01, rez02;

vreme=60*n+m;
memorija=16*vreme;
rezultat=k*1024-memorija;
rez01=rezultat/1024;
rez02=rezultat%1024;
cout << rez01 << " " << rez02 << endl;
return 0;
}
```

УЛАЗ	ИЗЛАЗ
1 2 3	2 32
100 34 4096	4001 736
8 56 12	3 640
1 1 1	0 48
14 56 14	0 0
1 26 3	1 672
4 0 4	0 256

3. Организује се аудиција за избор глумаца за нови серијал Ратова звезда. Пријавило се n глумаца. Познато је да је међу њима има a глумаца који су високи, b глумаца који су плавооки и c глумаца који су коврцави. За главну улогу у филму потребан је само један глумац - висок, плавих очију и коврцав. Да би испланирали време за појединачне интервјуе са глумцима, режисер мора да зна колики је максимални или минимални број пријављених глумаца који су погодни за ову улогу. Напишите програм AUDICIJA, који одређује максимални или минимални број глумаца са којима режисер мора да разговара. У првом реду стандардног улаза налази се један цео број који представља или максимални или минимални број глумаца који треба да се нађе. Овај број може имати следеће вредности:

1, ако је потребно да се утврди минимални број учесника;

2, ако је потребно да се утврди максималан број учесника.

У другом реду стандардног улаза налази се четири цела броја - n, a, b, c ($1 \leq n \leq 10000, 0 \leq a, b, c \leq n$) који су међусобно раздвојени једним бланко карактером.

У једином реду стандардног излаза програм треба да испише један цео број – минимални или максимални број глумаца (у зависност од датог улазног броја 1 или 2), који могу да освоје главну улогу.

Пример 1

УЛАЗ

2

5 3 4 5

ИЗЛАЗ

3

Пример 2

УЛАЗ

1

5 3 4 5

ИЗЛАЗ

2

Објашњење примера 1: Укупан број високих глумаца је 3, те на интервју не може ићи више од 3 глумца.

Објашњење примера 2: Сви глумци су коврцави и сви глумци осим једног су плавооки. Дакле међу три висока глумца су или 3 плавоока или 2 плавоока. Дакле имамо најмање два плавоока глумца међу високим глумцима (који су коврцави, наравно). Отуда, најмање двоје глумаца се интервјуишу за главну улогу.

РЕШЕЊЕ:

Овај задатак можемо поделити на два мања посебна задатка. Починимо са посебним задатком у ком се тражи максимални број глумаца. Тада је у питању случај да тражимо што је могуће више глумаца који су у истовреме високи, плавих очију и коврџави. Без ограничења општости, можемо да сматрамо да низ таквих глумаца почиње са глумцем број 1. Идеју можемо представити следећом шемом:

	1				b		a	c	n
Високи:									
Плавооки:									
Коврџави:									

Дакле, број максималних пресека кандидата ће бити једнак минимуму ова три скупа.

Погледајмо посебан задатак у ком се тражи минимални број глумаца. Најпре ћемо наћи пресек скупа високих и плавооких. Потом ћемо за тако добијени скуп наћи пресек са скупом коврџавих.

Пресек скупа високих и плавооких је приказан у табели. Опет, без смањења општости, почињемо са првим глумцем.

	1								n
Високи:									
Плавооки:									

	1				b		a	c	n
Високи									
плавооки:									
Коврџави:									

Пресек скупа високих и плавооких глумаца је пренесен у другу табелу и нумерација почиње са првим глумцем. Отуда можемо да додамо скуп коврџавих и поново задамо нумерацију глумаца и можемо да извршимо пренумерацију као у првој табели.

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

int main()
{int k, n, a, b, c,d,m,t;
cin>>k>>n;
cin>>a>>b>>c;
if(k==2)
{ if (b>c) d=c;    else d=b; //d je min (plavookih i kovrdzavih)
  if (d>a) m=a;    else m=d; // d je min (visokih, d) = min (visokih, plavookih, kovrdzavih)
}
else
{ if ( a+b-n>0) t=a+b-n;    else t=0; // skup visokih+plavooki mora veci od n
  if (c-n+t>0) c=c-n+t;    else c=0;
  if (t>c) m=c;    else m=t;  m=min(c,t)
}
cout<<m<<endl;
return 0;
```

}

УЛАЗ	ИЗЛАЗ
1 9 7 5 8	2
2 9 7 5 8	5
1 6 5 3 0	0
2 600 530 312 400	312
2 6000 5300 3142 470	470
1 6000 5300 3142 470	0
1 10000 10000 10000 10000	10000

**Општинско такмичење из програмирања за ученике основних школа
21. фебруар 2016.**

II категорија (7. и 8. разред)

1. исти као 2. задатак за 1. категорију

2. Освануо је и тај дан – чувени Концерт године на ком свирају напопуларнији бендови Србије. Концерт је почео у тачно h часова m минута и s секунди и бендови су свирали укупно k минута. Током концерта бендови су направили t пауза у трајању од по 5 минута. Напишите програм **KONCERT** који ће на стандардни излаз исписати када се завршио концерт. Са стандардног улаза се учитавају три цела броја h, m, s који означавају време почетка концерта ($8 \leq h \leq 23$, $0 \leq m \leq 59$, $0 \leq s \leq 59$), цео број k који означава колико минута је трајао концерт ($5 \leq k \leq 240$), цео број t који означава број пауза ($0 \leq t \leq 12$). На стандардном излазу исписати тачно време завршетка концерта у формату $hh:mm:ss$ где hh , mm , ss је редом време завршетка концерта. Водити рачуна о испису бројева са водећим нулама.

УЛАЗ	ИЗЛАЗ
20 00 00 60 4	21:20:00

Појашњење тест примера: Време почетка концерта је 20 ч 00 минута 00 секунди. Музичирање је трајало 60 минута, а паузе су трајале $4 \cdot 5 = 20$ минута. Зато се концерт завршио у 21:20:00

РЕШЕЊЕ:

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

int main()
{ int h,m,s, k, t, cas, minut;
  long long rezultat;
  cin >> h >> m >> s >> k >> t;
  rezultat=h*3600+m*60+s+k*60+t*5*60;
  cas=rezultat/3600%24;
  minut=(rezultat%3600)/60;
  cout << setfill('0') << setw(2) << cas << ":" << setfill('0') << setw(2) << minut << ":" << setfill('0') << setw(2)
  << s << endl;
```

```

return 0;
}

```

УЛАЗ	ИЗЛАЗ
20 00 00 60 4	21:20:00
20 01 02 180 4	23:21:02
21 10 00 180 4	00:30:00
20 00 35 184 5	23:29:35
22 10 11 143 7	01:08:11
08 30 00 178 7	12:03:00
08 59 59 181 7	12:35:59
10 00 00 120 12	13:00:00

3.

Дата су два природна броја a и b ($1 < a < b < 1000$). Растављање броја на просте чиниоце је поступак којим се неки природан број представља у облику производа чинилаца који су сви прости бројеви. Написати програм **RASTAVI**, који ће пребројати укупан број простих чинилаца на које се могу раставити природни бројеви који су између a и b (укључујући a и b). У једином реду стандардног улаза дата су два природна броја a и b међусобно раздвојена бланко карактером. У једином реду стандардног излаза исписати резултат.

ПРИМЕР

УЛАЗ	ИЗЛАЗ
7 10	8
17 17	1

Објашњење 1. тест примера. Растављањем на чиниоце, бројеви од 7 до 10 се могу написати у облику производа:

$$7 = 7$$

$$8 = 2 * 2 * 2$$

$$9 = 3 * 3$$

$$10 = 2 * 5$$

Укупан број свих простих чинилаца је 8.

РЕШЕЊЕ:

```

#include <iostream>
using namespace std;
int c;
void faktorisi(int n)
{ int d=2;
  while(n>1)
    if(n%d==0) {n=n/d; c++;}
    else d++;
}

int main()
{ int a,b;
  cin >> a >> b;
  for (int i=a;i<=b;i++) faktorisi(i);
  cout << c << endl;
  return 0;
}

```

ПРИМЕРИ

УЛАЗ	ИЗЛАЗ
2 3	2
10 11	3
13 133	308
2 201	514
11 999	2856
2 999	2871
900 999	301