

Домаћи задатак 001-004 (шампиони из 1д)

1.

Јована има m новчаница од 50 динара и двоструко више новчаница од 20 динара. Напиши програм који одређује колико јој новца остаје када купи n чоколадних бананица од којих свака кошта 15 динара (претпостави слободно да Јована има довољно новца да купи све бананице које жели). Са улаза се учитавају цели бројеви m и n (сваки у посебном реду, $m < 100$, $n < 100$).

Пример 1

Улаз:

3

10

Излаз:

120

Пример 2

Улаз:

2

6

Излаз:

90

```
#include <iostream>
using namespace std;
int main() {
    int m, n;
    cin >> m >> n;
    int novac = m*50 + (2*m)*20;
    int kusur = novac - n*15;

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

2. Стефанов брат је две године старији од њега, док је сестра три године млађа од њега. Напиши програм STEFAN који на основу Стефанових година израчунава збир година њих троје.

Улаз:

7

Излаз:

20

Анализа: Након уноса Стефанових година, потребно је израчунати колико година има Стефанов брат и колико година има Стефанова сестра. На крају се исписује збир Стефанових, братових и сестриних година.

```
#include <stdio.h>
int main()
{
int stefan, brat, sestra;
scanf("%d", &stefan);
brat = stefan + 2;
sestra = stefan - 3;

printf("%d\n", stefan + brat + sestra);
return 0;
}
```

3. Написати програм AVION који са улаза учитава у прва два реда време полетања авиона (у првом реду број сати између 0 и 23, у другом реду број минута између 0 и 59), а у следећа два реда време слетања авиона (у трећем реду број сати између 0 и 23, у четвртом реду број минута између 0 и 59) и исписује трајање лета у часовима и минутима. Претпоставити да су полетање и слетање у истом дану, као и да су све вредности исправно унете.

Улаз: 17

12

18

11

Излаз:

0 59

Улаз:

16

59

18

02

Излаз:

1 3

Анализа:

1. начин Израчунамо колико је секунди протекло од поноћи до почетка полетања и колико секунди је протекло од поноћи до почетка полетања. Одузмемо та два броја и добијемо трајање лета у секундама. Потом испишемо трајање лета у целим сатима и целим минутима.

2. начин Израчунамо колико је минута протекло од поноћи до почетка полетања и колико минута је протекло од поноћи до почетка полетања.

Одузмемо та два броја и добијемо трајање лета у секундама. Потом испишемо трајање лета у целим сатима и целим минутима.

```
#include <iostream>
using namespace std;
int main()
{
int poletanjeSat, poletanjeMinut, sletanjeSat,
sletanjeMinut, poletanje, sletanje, trajanjeLeta;
cin >> poletanjeSat >> poletanjeMinut;
cin >> sletanjeSat >> sletanjeMinut;
//pretvoriti vreme poletanja u sekunde
poletanje=poletanjeSat*3600+poletanjeMinut*60;
//pretvoriti vreme sletanja u sekunde
sletanje=sletanjeSat*3600+sletanjeMinut*60;
//izracunati trajanje leta u sekundama
trajanjeLeta=sletanje-poletanje;
//izdvajamo broj sati i broj minuta trajanja leta
int brojSati = trajanjeLeta/3600;
int brojMinuta = (trajanjeLeta %3600)/60;
cout << brojSati << " " << brojMinuta << endl;
return 0;
}
```

4. Наставница информатике жели да подели бомбоне својим ученицима. Напиши програм SLATKISI који учитава број бомбона које наставница има и број ученика и испишује колико најмање бомбона треба да купи да би сваки ученик добио исти број бомбона.

Улаз:

234

24

Излаз:

6

```
#include <iostream>
using namespace std;
int main() {
int broj_bombona, broj_ucenika;
cin >> broj_bombona >> broj_ucenika;
cout << (broj_ucenika - broj_bombona % broj_ucenika) %broj_ucenika << endl;
return 0;
}
```

5. Porodica je krenula autom na letovanje. Automobilom treba da pređu s kilometara krećući se ravnomernom brzinom od vkm/h. Napisati program kojim se određuje koliko će im sati trebati da pređu put.

Улаз

У првој линији standardnog ulaza се налази realna vrednost v , а у sledećој линији realna vrednost s које редом представљају brzinu изражену у km/h и planirani pređeni put у kilometrima.

Izlaz

Jedan realan broj zaokružen na dve decimale koji predstavlja potrebno vreme u satima.

Primer

Ulaz

60

1050

Izlaz

17.50

Rešenje

Brzina (v) pri ravnomernom kretanju jednaka je količniku pređenog puta (s) i vremena (t), tj. pri ravnomernom kretanju važi veza $v=s/t$,

pod pretpostavkom da se koriste odgovarajuće jedinice mere. Zato se vreme izračunava kao količnik pređenog puta i brzine kretanja (formula za vreme prilikom ravnomernog kretanja $t=s/v$). Prema tom, potrebno je ispisati količnik dva učitana broja.

```
#include <iostream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    double v, s;
```

```
    cin >> v >> s;
```

```
    double t = s / v;
```

```
    cout << fixed << showpoint << setprecision(2) << t << endl;
```

```
    return 0;
```

```
}
```

6. Sa istog startnog mesta kreću dva automobila prvi se kreće brzinom v_1 m/s, drugi v_2 m/s pri čemu je $v_2 > v_1$. Drugi automobil polazi t minuta posle prvog. Napisati program kojim se određuje posle koliko minuta od polaska drugi automobil sustiže prvog.

Ulaz

Unose se tri realna broja koja redom predstavljaju brzine prvog i drugog automobila izražene u m/s i broj minuta posle koliko kreće drugi automobil. Svaki broj je u posebnoj liniji

standardnog ulaza.

Izlaz

Na standarnom izlazu prikazati broj minuta zaokružen na dve decimale posle koliko će drugi automobil sustiže prvog.

Primer

Ulaz

20

25

10

Izlaz

40.00

Rešenje :

Automobili se kreću ravnomerno, tako da će se rešenje zasnivati na vezi između brzine, pređenog puta i vremena $v=s/t$.

Automobili polaze sa istog mesta pa do sustizanja dolazi u momentu kada automobili pređu isti put. Traženo vreme možemo odrediti izjednačavanjem pređenih puteva automobila.

Za početak potrebno je da sve izrazimo u istim jedinicama. Ako je brzina automobila jednaka v (m/s), pošto jedan minut iznosi 60 sekundi (tj. važi da je $\text{min}=60\text{s}$), tada je brzina jednaka $v_{\text{m/s}} \cdot 60_{\text{s/min}} = (v \cdot 60)_{\text{m/min}}$.

Ako obeležimo sa x broj minuta posle koliko drugi automobil sustiže prvog. Prvi automobil se kreće $t+x$ minuta a drugi x minuta. Pređeni put pri ravnomernom kretanju jednak je proizvodu brzine i vremena ($s=v \cdot t$). Zato je put koji do sustizanja pređe prvi automobil

$$(t+x)\text{min} \cdot ((v_1 \cdot 60)_{\text{mmin}}) = (t+x) \cdot v_1 \cdot 60\text{m},$$

a put koji do sustizanja pređe drugi automobil

$$x\text{min} \cdot ((v_2 \cdot 60)_{\text{mmin}}) = x \cdot v_2 \cdot 60\text{m}.$$

Pošto su u trenutku susreta oba automobila prešla isto rastojanje, izjednačavanjem pređenih puteva dobijamo jednačinu $(t+x) \cdot 60 \cdot v_1 = x \cdot 60 \cdot v_2$, tj. $(t+x) \cdot v_1 = x \cdot v_2$ čije rešenje je

$$x = v_1 \cdot t v_2 - v_1.$$

```
#include <iostream>
```

```
#include <iomanip>
```

```
using namespace std;
```

```
int main() {
```

```
double v1, v2, t;
```

```
cin >> v1 >> v2 >> t;
```

```
double x = (v1 * t) / (v2 - v1);
```

```
cout << fixed << showpoint << setprecision(2) << x << endl;
```

```
return 0;
```

```
}
```

7. Na planeti M u ulici M žive mravi. Sve kućice u ulici su označene prirodnim brojevima. Mravica Ana živi u kućici broj a , a mrav Boris iza nje, u kućici

broj b. Mrav Viktor živi u kućici tačno na sredini između Ane i Borisa. Napiši program koji određuje broj kućice u kojoj živi Viktor (ako je između Ane i Borisa paran broj kućica, Viktor živi u onoj koja je bliža Ani).

Ulaz

Sa standardnog ulaza unose se dva cela broja:

•a ($0 \leq a \leq 2 \cdot 10^9$) - broj kućice u kojoj živi Ana

•b ($a+2 \leq b \leq 2 \cdot 10^9$) - broj kućice u kojoj živi Boris

Izlaz

Na standardni izlaz ispisati jedan ceo broj - broj kućice u kojoj živi Viktor.

Primer 1

Ulaz

432457

562321

Izlaz

497389

Primer 2

Ulaz

2

9

Izlaz

5

Primer 3

Ulaz

2000000000

1999999998

Izlaz

1999999999

Rešenje 1:

```
#include <iostream>
```

```
using namespace std;
```

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

Rešenje 2:

```
#include <iostream>
```

```
using namespace std;
```

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

8. Na šahovskoj tabli širine n i dužine m , gornje levo polje je bele boje. Napisati program kojim se određuje broj crnih polja na datoj šahovskoj tabli.

Ulaz

Na standardnom ulazu nalaze se dva prirodna broja $n, m \leq 100$.

Izlaz

Broj crnih polja na šahovskoj tabli.

Primer

Ulaz

5

3

Izlaz

7

Rešenje

Na šahovskoj tabli dimenzije $n \times m$ ukupan broj polja je $n \cdot m$. U svakom redu boja polja se naizmenično menja. Redovi počinju naizmenično belom pa crnom bojom (prvi red počinje belom bojom, drugi crnom bojom, treći red belom, ...).

Ako je ukupan broj polja paran onda crnih i belih polja ima jednak broj, a ako je ukupan broj polja neparan broj belih polja je za 1 veći od broja crnih polja (jer prvi red počinje belim poljem). Prema tome broj crnih polja jednak je celobrojnom količniku ukupnog broja polja i broja 2.

U programskim jezicima C i C++, operator $/$ izračunava celobrojni količnik kada su oba operanda celi brojevi.

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
int n, m;  
cin >> n >> m;  
int brCrnihPolja = (n * m) / 2;  
cout << brCrnihPolja << endl;  
return 0;  
}
```

9. Na šahovskoj tabli 8×8 šahovska polja se broje, počev od donjeg levog polja, kolona po kolona od A1 do H8. Napisati program kojim se određuje u kojoj koloni i

kom redu je polje koje je n -to izbrojano i koje po redu će biti izabrano polje u datoj koloni a i redu b .

Ulaz

Prva linija standardnog ulaza sadrži prirodan broj $1 \leq n \leq 64$, druga linija sadrži dva karaktera, prvo slovo od **A** do **H** koje predstavlja kolonu a , a zatim cifru od **1** do **8** koja predstavlja red b .

Izlaz

U prvoj liniji standardnog izlaza prikazati broj kolone (slovo od **A** do **H**) i broj reda (cifru od **1** do **8**) polja šahovske table koje je n -to izbrojano. Druga linija standardnog izlaza sadrži redni broj polja koje se nalazi u koloni a i redu b .

Primer

Ulaz

19

F5

Izlaz

C3

45

Rešenje

Prikažimo prvo kako se može rešiti drugi deo zadatka. Određivanje rednog broja polja lakše je ako bi brojanje polja i brojanje kolona i redova kretalo od nule. Zaista, razmotrimo da su polja numerisana na sledeći način:

7 15 23 31 39 47 55 63 ← 7

6 14 22 30 38 46 54 62 ← 6

5 13 21 29 37 45 53 61 ← 5

4 12 20 28 36 44 52 60 ← 4

3 11 19 27 35 43 51 59 ← 3

2 10 18 26 34 42 50 58 ← 2

1 9 17 25 33 41 49 57 ← 1

0 8 16 24 32 40 48 56 ← 0

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

0 1 2 3 4 5 6 7

Ako se traži broj u koloni k i redu r ispred njega postoji tačno k kolona sa po 8 elemenata, a u tekućoj koloni postoji tačno r elemenata ispred njega. Dakle, taj broj se može odrediti izrazom $8 \cdot k + r$. Na primer, ako računamo broj elementa u koloni broj 3 i redu broj 3, to će biti broj $8 \cdot 3 + 3$

tj. Broj 27

Prisetimo da u ovom slučaju zadatak odgovara određivanju vrednosti oktalnog zapisa $(kr)_8$

Ostaje pitanje kako na osnovu oznake kolone i reda (koje su predstavljene sa dva karaktera koje ćemo označiti sa a i b) dobiti njihove redne brojeve k i r . Redni broj

kolone k se može izračunati tako što se od ASCII koda karaktera koji označava kolonu oduzme ASCII kod karaktera 'A' tj. izrazom $a - 'A'$ (za oznaku 'A' to će biti 0, za oznaku 'B' to će biti 1 itd.). Redni broj reda r se može dobiti tako što se od ASCII koda karaktera koji označava red oduzme ASCII kod karaktera '1' tj. izrazom $b - '1'$ (za oznaku '1' to će biti 0, za oznaku '2' to će biti 1 itd.). Na kraju, pošto se rezultujući broj traži u intervalu od 1 do 64, a ne od 0 do 63, na dobijeni rezultat potrebno je dodati 1. Stoga se redni broj elementa može dobiti izrazom $(a - 'A') * 8 + (b - '1') + 1$. Prvi deo zadatka se može rešiti tako što se odredi inverzna transformacija ovoj iz drugog dela zadatka.

Pošto je vrednost izraza $b - '1'$ u intervalu od 0 do 7, ona se može iz rezultat zračunati kao ostatak pri deljenju vrednosti $n - 1$ brojem 8.

Otuda se oznaka reda b može izračunati izrazom $(n - 1) \% 8 + '1'$.

Slično, vrednost izraza $a - 'A'$ se može izračunati kao celobrojni količnik pri deljenju vrednosti $n - 1$ sa 8, pa se oznaka kolone a može dobiti izrazom $(n - 1) / 8 + 'A'$.

```
#include <iostream>

using namespace std;

int main() {
// redni broj polja
int n;
cin >> n;
// odredjujemo oznake kolone i reda sa rednim brojem n
char a = (n - 1) / 8 + 'A';
char b = (n - 1) % 8 + '1';
cout << a << b << endl;

// učitavamo oznake novog polja
cin >> a >> b;
// odredjujemo njegov redni broj
n = (a - 'A') * 8 + (b - '1') + 1;
cout << n << endl;

return 0;
}
```

10. **Podela jabuka.** Aca je dobio n jabuka i na jednake delove (po trećinu) podelio braći Branku, Canetu, i Dušanu. Preostale jabuke pojeo je sam. Branko je svoje jabuke podelio braći (Aci, Canetu i Dušanu), a što je preostalo pojeo je sam. Tako su postupili prvo Cane, pa onda i Dušan. Koliko jabuka posle ovakve podele ima svako od njih (zanemarujući one koje su pojedene).

Ulaz

Broj jabuka n koje je dobio Aca.

Izlaz

Broj jabuka koje je dobio Aca (prvi red), Branko (drugi red), Cane (treći red) i Dušan (četvrti red).

Primer

Ulaz

100

Izlaz

44

33

19

0

Objašnjenje

Aca je dobio 100 jabuka. Po 33 je dao Branku, Canetu i Dušanu, a jednu je pojeo sam, tako da su nakon prve podele Aca, Branko, Cane i Dušan imali redom 0, 33, 33, 33 jabuke. Nakon toga, Branko je podelio braći po 11 jabuka i ništa mu nije ostalo, tako da je broj jabuka nakon druge podele 11, 0, 44, 44. Nakon toga, Cane je delio jabuke tako što je braći dao po 14 jabuka, a dve je sam pojeo, tako da je nakon treće podele stanje 25, 14, 0, 58. Na kraju, Dušan je braći dao po 19 jabuka, a jednu preostalu je sam pojeo. Zato je na kraju stanje bilo 44, 33, 19, 0.

Rešenje

Osnovni korak podele (koji se ponavlja 4 puta) je da neka osoba koja pre tog koraka ima x jabuka, daje trećinu svojih jabuka osobama koje imaju y , z i w jabuka. Nakon tog koraka podele osobe čija je količina jabuka bila y , z i w uvećavaju svoju količinu jabuka za celobrojni količnik broja x sa 3 tj. $x \div 3$, dok osobi čija je količina jabuka bila x i koja deli jabuke ostaje 0 (što je nova vrednost za x), jer sve jabuke ili podeli ili pojede.

U prvom koraku podele osoba x je Aca, dok su ostale tri osobe (y , z i w) Branko, Cane i Dušan, u drugom je osoba x Branko, dok su ostale tri osobe Aca, Cane i Dušan i tako dalje.

Na početku Aca ima dati broj jabuka n , dok svi ostali imaju 0 jabuka. Ako napravimo potprogram koja realizuje jedan korak podele, taj potprogram možemo pozvati četiri puta i to tako što u prvom pozivu parametar x vezujemo za Acin broj jabuka, u drugom za Brankov, u trećem za Canetov, a u četvrtom za Dušanov, dok se ostali parametri (y , z i w) vezuju za broj jabuka ostale braće.

11. **Za date prirodne brojeve a i b napisati program kojim se dati nepravilni razlomak a/b prevodi u mešoviti broj**

Ulaz

U prvoj liniji standardnog ulaza nalazi se prirodan broj a koji predstavlja brojilac nepravog razlomka, a u drugoj liniji prirodan broj b različit od nule koji predstavlja imenilac razlomka ($a \geq b$).

Izlaz

Prva i jedina linija standardnog izlaza sadrži mešoviti zapis razlomka, preciznije prirodan broj, brojilac i imenilac mešovitog broja međusobno odvojeni sa po jednom prazninom (blanko znakom).

Primer**Ulaz**

23

8

Izlaz

2 7 8

REŠENJE

```
#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;  
}
```

12. **Vaterpolista Marko učestvuje na pripremama za takmičenje u trajanju od n dana. Prvog dana priprema Marko prepliva a metara, a svakog sledećeg dana za d metara više nego prethodnog dana. Napisati program kojim se određuje koliko metara Marko prepliva poslednjeg dana priprema.**

Ulaz

Unose se tri prirodna broja, svaki u posebnoj liniji standardnog ulaza. Brojevi redom predstavljaju broj dana koliko pripreme traju n ($n \leq 20$), koliko metara Marko prepliva prvog dana priprema a ($a \leq 3000$), koliko metara svaki dan Marko više pliva u odnosu na prethodni dan d ($d \leq 1000$).

Izlaz

Na standardnom izlazu prikazati koliko metara Marko prepliva poslednjeg dana priprema.

Primer**Ulaz**

10

500

100

Izlaz

1400

Rešenje

Marko prvog dana prepliva a metara. Drugog dana prepliva za d metara više nego prvog dana, znači $a+d$ metara. Trećeg dana prepliva za d metara više nego drugog dana što iznosi $a+2 \cdot d$ metara, itd. Možemo zaključiti da će poslednjeg dana priprema, n -tog dana, preplivati $a+(n-1) \cdot d$ metara.

Skrenimo pažnju da je u pitanju aritmetički niz tj. aritmetička progresija (niz brojeva u kome je razlika svakog člana i njemu prethodnog konstantna) i da je potrebno odrediti n-ti član aritmetičke progresije.

Još jedno moguće rešenje je da se upotrebi petlja u kojoj bi se za svaki naredni dan uvećavalo preplivano rastojanje za d (i to bi se izvršilo n-1 puta). Za veće vrednosti n, ovakvo rešenje je znatno neefikasnije nego polazno u kojem se koristi izvedena matematička formula.

```
#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;  
}
```

13. Сваки човек носи два хлеба, жена један, а дете пола хлеба. Укупно их је n, и носе n хлебова. Написати програм који учитава природан број n и исписује колико има решења за број људи, жена и деце.

Улаз

Са стандардног улаза се уноси број n ($1 \leq n \leq 10^9$).

Излаз

На стандардни излаз исписати број решења

Пример 1

Улаз

5

Излаз

2

Објашњење

Могућа решења су (1,2,2) и (0,5,0).

Пример 2

Улаз

12

Излаз

5

Rešenje

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {  
int n;  
cin >> n;  
cout << n / 3 + 1 << endl;  
return 0;
```

```
}
```

14. Колики је збир бројева у датом реду следећег троугла?

```
    1
   2 3 4
  5 6 7 8 9
10 11 12 13 14 15 16
...

```

Улаз: Редни број k ($1 \leq k \leq 5 \cdot 10^5$), реда троугла чији збир треба израчунати (бројање редова почиње од 1).

Излаз: Збир вредности у задатом реду троугла.

Пример

Улаз

3

Излаз

35

Rešenje:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
long long k;
cin >> k;
long long zbirRedaTrougla = (2 * k - 1) * (k * k - k + 1);
cout << zbirRedaTrougla << endl;
return 0;
}
```

15. Дати су цели бројеви n и k . Напиши програм који одређује највећи цео број $l \leq n$ дељив бројем k и најмањи цео број $d \geq n$ дељив бројем k .

Улаз: Са стандардног улаза уносе се два цела броја (сваки у посебном реду).

• n ($0 \leq n \leq 100000$)

• k ($1 \leq k \leq 10000$)

Излаз: На стандардни излаз исписати два тражена цела броја, сваки у посебном реду.

Пример 1

Улаз

23

5

Излаз

20

25

Пример 2

Улаз

49

7

Излаз

49

49

Решење:

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
int n, k;
```

```
cin >> n >> k;
```

```
int l = (n / k) * k; // floor(n/k) = n/k
```

```
int d = ((n + k - 1) / k) * k; // ceil(n/k) = (n+k-1)/k
```

```
cout << l << endl << d << endl;
```

```
return 0;
```

```
}
```