

Naredbe grananja

Zadaci za vežbu

NCP = Napisati C/C++ program

1. NCP koji će učitati Vašu godinu rođenja i ispisati na standardni izlaz da li ste rođeni u prestupnoj godini.

Rešenje

```
/* godina je prestupna ako je deljiva sa 4 i nije deljiva sa 100 ili je deljiva sa 400 */
/* UOCITE: if naredba se moze koristiti i bez else grane */
#include <stdio.h>
int main()
{ int g;
printf("Unesite godinu rođenja: ");
scanf("%d", &g);
if( (g%4==0 && g%100!=0) || (g%400==0) )
    printf("Rodjeni ste u prestupnoj godini!\n");
else printf("Niste rodjeni u prestupnoj godini!\n");
}
```

2. NCP koji učitava 1 karakter sa ulaza i ispituje da li je uneti karakter cifra ili veliko slovo ili malo slovo.

3. Dat je ceo broj K od 1 do 180. NCP koji će ispisati koja se cifra nalazi na K-toj poziciji niza 10111213...9899 u kom su redom ispisani brojevi od 10 do 99.

ULAZ	IZLAZ
1	1
2	0
3	1
6	2
22	0
101	6
179	9

Rešenje

```
#include <stdio.h>
main()
{
int k,b;
scanf("%d", &k);

b=10+(k-1)/2;
if ((k-1)%2==0) printf("%d", b/10);
else printf("%d", b%10);
}
```

4. NCP koji za dati redni broj meseca i godinu određuje broj dana u tom mesecu. Voditi računa o tome da li je godina prestupna (godina je prestupna ako je deljiva sa 4, a nije deljiva sa 100, osim ako je deljiva sa 400, kada jeste prestupna).

Ulaz

Sa standardnog ulaza učitavaju se u jednom redu dva broja razdvojena blanko karakterom: broj meseca m ($1 \leq m \leq 12$) i broj godine g ($1900 \leq g \leq 2100$)

Izlaz

Na standardni izlaz ispisati jedan ceo broj koji predstavlja broj dana u zadatom mesecu.

Primer

ULAZ	IZLAZ
12 2016	31
1 2016	31
2 2016	29
2 2017	28
5 2017	31

Rešenje

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

#define prestupna(godina) ((godina % 4 == 0 && godina % 100 != 0) || (godina % 400) == 0)
// provera da li je data godina prestupna

int main() {
    // učitavamo mesec i godinu
    int mesec, godina;
    cin >> mesec >> godina;

    // odredjujemo broj dana u tom mesecu
    int brojDana = 0;
    // januar, mart, maj, jul, avgust, oktobar, decembar
    if (mesec == 1 || mesec == 3 || mesec == 5 || mesec == 7 || mesec == 8 || mesec == 10 || mesec ==
12)
        brojDana = 31;
    // april, jun, septembar, novembar
    else if (mesec == 4 || mesec == 6 || mesec == 9 || mesec == 11)
        brojDana = 30;
    // februar
    else if (mesec == 2)
        brojDana = prestupna(godina) ? 29 : 28;

    // ispisujemo rezultat
    cout << brojDana << endl;
    return 0;
}
```

5. NCP koji učitava sa standardnog ulaza datum u formatu dd.mm.gggg i ispisuje na standardni izlaz datum prethodnog dana. Pretpostaviti da datumi su korektno uneti i da se odnose na vreme posle 15.10.1582.

Na primer

```
ulaz 01.11.2017. izlaz 31.10.2017.
ulaz 01.01.2017. izlaz 31.12.2016.
ulaz 01.03.2000. izlaz 29.02.2000.
ulaz 01.03.2001. izlaz 28.02.2001.
```

6. NCP koji učitava dva intervala realne prave i određuje dužinu najmanjeg intervala koji sadrži ta dva intervala.

```
ulaz          izlaz
(5,8) (6,9)   4
(-1,2) (3,4)  5
```

Rešenje

```
#include <stdio.h>
main() {
float x1,y1,x2,y2,rez=0; /*zadate tacke,duzina intervala pokrивaca*/
float gg,dg; /*gornja granica pokrивaca, donja granica pokrивaca*/

//printf("Unesite tacke u obliku (a,b)(c,d):\n");
scanf("(%f,%f)(%f,%f",&x1,&y1,&x2,&y2);

    gg=y1;
    dg=x2;
    if (x2>x1) dg=x1;
    if (y2>y1) gg=y2;
    rez= gg-dg;

printf("Duzina intervala koju pokriva intervale je %f", rez);
}
```

Zadaci za vežbu – priprema za takmičenje

1. U listi celih brojeva 1, 2, ..., N možete izvršiti niz cut-paste operacija. Kažemo da cut-paste operacija $\langle x, y, z \rangle$ se sastoji od preseka fragmenta između vrednosti x i y ubacivanja fragmenta odmah posle vrednosti z (važi da z može da bude 0 kako bi se omogućilo ubacivanje na početku). Uređena trojka $\langle x, y, z \rangle$ predstavlja ispravnu cut-paste operaciju ako

x se pojavljuje ispred y u listi, ili $x = y$;

z se pojavljuje van fragmenta od x do y , ili $z = 0$.

Odredite niz ispravnih operacija koje preokreću listu, tako da nakon izvršenih operacija, lista ima sadržaj redom $N, N - 1, \dots, 2, 1$. Više poena ćete osvojiti ukoliko odrediti što je moguće manje operacija.

Ulaz

Datoteka cpr.in sadrži jedan ceo broj N , broj članova liste.

Izlaz

Datoteka cpr.out u prvoj liniji sadrži jedan broj M čija vrednost je broj nađenih cut-paste operacija.

Svaka od narednih M linija mora da sadrži tri broja x y z koja predstavljaju operaciju.

Ograničenja

$1 \leq N \leq 5,000$

Vremensko ograničenje: 0.1 sekunda

Memorijsko ograničenje: 64 MB

Ako za svaki test primer Vaš program ispiše da je broj operacija jednak M onda ćete osvojiti poene prema dole navedenoj skali (kosa crta / označava celobrojno deljenje):

100% poena ako važi da je $M \leq N / 2 + 1$;

80% poena ako važi da je $N / 2 + 1 < M \leq 2 * N / 3$;

60% poena ako važi da je $2 * N / 3 < M \leq 3 * N / 4$;

40% poena ako važi da je $3 * N / 4 < M \leq 4 * N / 5$;

20% poena ako važi da je $4 * N / 5 < M \leq 5 * N / 6$;

0 poena ako važi da je $M > 5 * N / 6$.

Primer

ULAZ

6

IZLAZ

4

2 6 0

4 5 0

3 6 4

6 5 0

Pojašnjenje

Polazna lista je 1 2 3 4 5 6

Nakon prve operacije, lista glasi 2 3 4 5 6 1. Uočite da operacija (1 1 6) daje isti rezultat.

Nakon druge operacije, lista glasi 4 5 2 3 6 1

Nakon treće operacije, lista glasi 4 3 6 5 2 1

Nakon četvrte operacije, lista glasi 6 5 4 3 2 1

Ovo rešenje osvaja 100% poena jer važi da $4 = 6 / 2 + 1$.

MALA POMOĆ

Prođimo ručno kroz primere za $n=1,2,3,4,5,6,7,8,9,12$

cpr.in	cpr.out	Појашњење
1	0	
2	1 2 2 0	Broj 2 ide na pocetak
3	2 2 2 3 1 1 2	1 2 3 1 3 2 3 2 1
4	3 2 2 3 1 1 2 4 4 0	1 2 3 4 1 3 2 4 3 2 1 4 4 3 2 1

5	3 1 2 4 2 5 3 5 4 0	1 2 3 4 5 3 4 1 2 5 3 2 5 4 1 5 4 3 2 1
6	4 1 2 4 2 5 3 5 4 0 6 6 0	1 2 3 4 5 6 3 4 1 2 5 6 3 2 5 4 1 6 5 4 3 2 1 6 6 5 4 3 2 1
7	4 1 2 5 2 7 3 7 4 0 6 5 7	1 2 3 4 5 6 7 3 4 5 1 2 6 7 3 2 6 7 4 5 1 7 4 3 2 6 5 1 7 6 5 4 3 2 1
8	5 1 2 5 2 7 3 7 4 0 6 5 7 8 8 0	1 2 3 4 5 6 7 8 3 4 5 1 2 6 7 8 3 2 6 7 4 5 1 8 7 4 3 2 6 5 1 8 7 6 5 4 3 2 1 8 8 7 6 5 4 3 2 1
9	5 1 2 6 2 9 3 9 4 0 8 5 9 7 6 8	1 2 3 4 5 6 7 8 9 3 4 5 6 1 2 7 8 9 3 2 7 8 9 4 5 6 1 9 4 3 2 7 8 5 6 1 9 8 5 4 3 2 7 6 1 9 8 7 6 5 4 3 2 1
12	7 1 2 7 2 11 3 11 4 0 10 5 11 9 6 10 8 7 9 12 12 0	1 2 3 4 5 6 7 8 9 10 11 12 3 4 5 6 7 1 2 8 9 10 11 12 3 2 8 9 10 11 4 5 6 7 1 12 11 4 3 2 8 9 10 5 6 7 1 12 11 10 5 4 3 2 8 9 6 7 1 12 11 10 9 6 5 4 3 2 8 7 1 12 11 10 9 8 7 6 5 4 3 2 1 12 12 11 10 9 8 7 6 5 4 3 2 1

2.

<http://mendo.mk/Task.do?id=691>

Programeri

Bojan je rešio da oformi Partiju programera (PP) i da učestvuje na predstojećim izborima. Da bi je registrovao, treba mu spisak članova. Pošto ima kontakt od svih N programera, šalje im poziv da se učlane u PP. Oni odgovaraju da prihvataju ili odbijaju. Bojan, zatim pravi spisak novih članova. Želi da spisak bude što veći, te ponavlja postupak. Ponovo šalje poziv svima uključujući trenutni spisak. Programer će prihvatiti članstvo ako je bar jedan njegov/njen prijatelj na spisku. U suprotnom će odbiti članstvo. Posle druge runde, Bojan pravi novi spisak i šalje treći put.....Ako Bojan zna koji programer se s kim druži, pomoz mu da izračuna broj programera koji će biti na najdužoj listi (sačinjenoj nakon prvog, drugog puta)

Ulaz

Prva linija sadrži N , broj programera ($1 \leq N \leq 50000$). Druga linija sadrži N celih brojeva B_i razdvojenih jednim blanko karakterom, $1 \leq i \leq N$, gde je $B_i=1$ ako je i -ti programer prihvatio prvi poziv za članstvo i $B_i=0$ ako je odbio. Treća linija sadrži M , broj parova prijatelja ($0 \leq M \leq 200000$). U svakoj od narednih M linija, nalazi se par celih brojeva koji predstavljaju par programera koji su prijatelji.

Izlaz

Izlaz se sastoji od jednog celog broja – broj programera na najdužoj listi.

Ograničenja

Vremensko ograničenje: 1 sekunda

Memorijsko ograničenje: 64 MB

Primeri

```
ulaz          izlaz
3             2
1 0 0
2
1 2
1 3
```

```
ulaz          izlaz
3             3
1 1 0
2
1 2
1 3
```

Objašnjenje prvog primera: Samo je prvi programer prihvatio prvi poziv. Onda je Bojan poslao spisak svima. Posle drugog poziva, Programeri 2 i 3 su prihvatili (zato što je 2 prijatelj sa 1 i 3 je prijatelj sa 1 – koji je bio na prvom spisku), a programer 1 odbija članstvo jer ni 2 ni 3 nisu bili na prvom spisku. Spisak programera koji su prihvatili članstvo nakon druge runde su Programer 2 i Programer 3. Posle sledeće runde poziva samo će Programer 1 biti na spisku.... itd. Dakle, broj programera na najdužem spisku je 2.

Objašnjenje drugog primera: Posle prvog poziva 1 i 2 su na spisku. U drugoj rundi 1 će prihvatiti jer je prijatelj sa 2 koji je na spisku, 2 će prihvatiti jer se njegov prijatelj 1 nalazi na spisku i 3 će prihvatiti jer je njegov prijatelj 1 na spisku. Sva 3 programera su na spisku, dakle odgovor je 3.

Rešenje

Koristićemo neusmereni graf, čiji čvorovi su programeri prijatelji i grane povezuju parove prijatelja. U početku, markirani su čvorovi koji odgovaraju prijateljima koji su dali svoj pristanak. Dalje, svaka iteracija (ponavljanje) stvara novu listu označenih čvorova: u listu su uključeni oni i samo oni čvorovi koji su u prethodnom koraku imali najmanje jednog markiranog suseda u grafu.

Razmotrimo ključne karakteristike tokom postupka izmene i dopune liste (spiska) označenih čvorova.

Neka u tekućem koraku imamo dva susedna temena, od kojih je najmanje jedan označen. Onda u sledećoj iteraciji čvor koji nije u partiji može da "izađe", ali nakon naredne iteracije - on može opet biti markiran. Napominjemo da ovaj čvor će redovno odlaziti/ulaziti na listu markiranih čvorova nakon svakih dva ponavljanja.

Ako su markirana dva susedna čvora, oni će ostati markirani, bez obzira na broj iteracija.

Uzimajući u obzir ove zakone predlažemo sledeći algoritam za rešavanje problema. Bojimo graf u dve boje koristeći BFS: prvi čvor svake komponente povezanosti bojimo u belo i sve njegove susede - u crnu boju, sve susedne komponente - opet belo i tako dalje .. Ako pri tom bilo koja komponenta povezanosti postoji ciklus neparne dužine, biće kolizij, komponenta povezanosti ne može biti obojena u dve boje, tako da sva susedna temena belog čvora budu crna, i svi čvorovi susedni sa crnim da bude beli .

