Naredba switch

switch(iskaz) {
case konst1: iskaz1; break;
case konst2: iskaz2; break;
case konst100: iskaz100; break;
default: iskaz101;
}
	if (iskaz==konst1) iskaz1;
	else if (iskaz==konst2) iskaz2;
	
	else if (iskaz==konst100) iskaz100;
	else iskaz 101;

Primer1:

switch(mesto) {
case 1: printf("\nZlato"); break;
case 2: printf("\nSrebro"); break;
case 3: printf("\nBronza"); break;
default: printf("\nBez medalje");
}
	if (mesto==1) printf("\nZlato");
	else if (mesto==2) printf("\nSrebro");
	else if (mesto==3) printf("\nBronza");
	else printf("\nBez medalje");

switch(iskaz) {
case konst1:
case konst2:
case konst100: iskaz100; break;
default: iskaz101;
}
	if ((iskaz==konst1) ||(iskaz==konst2) ||
	(iskaz==konst100))
	iskaz100;
	else iskaz101;

Primer2:

switch(mesto) {
case 1: case 2: case 3:
printf("\nMedalja"); break;
default: printf("\nBez medalje");
}
	if ((mesto==1) ||(mesto==2) ||(mesto==3))
	printf("\nMedalja");
	else printf("\nBez medalje");

1. NCP koji učitava dva cela broja i koristeći switch naredbu ispisuje njihov maksimum.

```
#include <stdio.h>
main()
{
  int a,b, max;
scanf("%d%d",&a,&b);
switch(a>b)
{
    case 0:max=b;break;
    case 1:max=a;
  }
   printf("\nmax=%d\n",max);
}
```

2. NCP koji učitava malo slovo, a na standardni izlaz ispisuje da li je učitano slovo samoglasnik a,e,i,o,u.

```
#include <stdio.h>
main()
{
  char x;
scanf("%c",&x);
switch(x)
{
    case 'a': case 'e': case 'i': case 'o': case 'u':
        printf("%c je samoglasnik\n",x); break;
    case 'r': printf("slovo r\n"); break;
     default: printf("%c je suglasnik\n",x);
  }
}
```

3. SWITCH – primer 3, jednostavni kalkulator

Napisati C program koji učitava komandu sa ulaza (u vidu jednog karaktera) i na osnovu te komande učitava dva broja nad kojima izvršava odgovarajuću operaciju. Ovaj postupak se ponavlja dok se ne unese komanda za kraj programa.
Program ilustruje primenu do-while petlje i switch naredbe za implementaciju menija, za potrebe interaktivnog rada.

Resenje

```
#include <stdio.h>
int main() {
char c;   int a,b;

/* Naredna do-while petlja predstavlja tipican nacin za implementaciju interaktivnog menija: u svakoj iteraciji se
ispisuje meni, a od nas se ocekuje da unesemo karakter za komandu.
U zavisnosti od unetog karaktera se preduzima odgovarajuca akcija (u ovom slucaju unos dva broja i prikaz
rezultata odgovarajuce operacije).
Petlja se izvrsava sve dok se ne unese komanda za kraj ('q' u ovom slucaju).*/


do {          /* Prikaz menija */
printf("Unesite:\n");    printf("\t\'a\' za sabiranje,\n\t\'s\' za oduzimanje,\n")
printf("\t\'m\' za mnozenje,\n\t\'d\' za deljenje,\n");  printf("\t\'q\' za kraj.\n");
```

```
scanf("%c", &c); /* Ucitavamo komandu */

/* Izbor akcije na osnovu komande koja je uneta */

switch(c) {
case 'a':     /* Slucaj sabiranja */
printf("Unesite dva cela broja: "); scanf("%d%d", &a, &b);
printf("%d+%d=%d\n", a, b, a + b);  break;

/* Slucaj oduzimanja */
case 's':
printf("Unesite dva cela broja: ");  scanf("%d%d", &a, &b);
printf("%d-%d=%d\n", a, b, a – b);
break;
/* Slucaj mnozenja */
case 'm':
printf("Unesite dva cela broja: "); scanf("%d%d", &a, &b);
printf("%d*%d=%d\n", a, b, a *b); break;
/* Slucaj deljenja */
case 'd':
printf("Unesite dva cela broja: "); scanf("%d%d", &a, &b);
printf("%d/%d=%d\n", a, b, a / b);break;
/* Slucaj izlaska -- ne radimo nista */
case 'q': break;
/* Slucaj pogresne komande(sve ostalo sto nije navedeno gore) */
default: printf("Pogresna komanda!\n"); break;
}
} while (c != 'q');        /* Iz petlje izlazimo kada je c == 'q' */
return 0;
}
```

4. Napisati C program koji učitava sa standardnog ulaza datum u formatu dd.mm.gggg i ispisuje na standardni izlaz redni broj tog dana u godini. Pretpostaviti da datumi su korektno uneti i da se odnose na vreme posle 15.10.1582.
Na primer ulaz 01.01.2012. izlaz 1
ulaz 03.02.2011. izlaz 34
ulaz 01.03.2012. izlaz 61
ulaz 01.03.2011. izlaz 60

```
#include <stdio.h>
main(){
   int d,m,g; /*ucitani dan, mesec, godina*/
   short gp; /*indikator da li je godina prestupna*/
   int k,m1; /*redni broj dana, redni broj meseca koji prethode mesecu m*/
   int br; /* broj dana meseca m1 godine g*/
scanf("%d.%d.%d.",&d,&m,&g);
m1=1;k=0;

while (m>m1){
  switch(m1){
   case 4:case 6:case 9:case 11:br=30;break;
   case 2:gp=((g%4==0) && (g%100!=0)) || (g%400==0); br=28+gp;break;
   default:br=31;
  }
 k+=br;
 m1++;
}
```

```
k+=d;
printf("%d",k);
}
```

Korisni formati učitavanja

5. Šta je rezultat rada sledećeg fragmenta programa ako imamo
ULAZ
pera mika laza toma
12.4567 85   2.12345678

```
#include <stdio.h>
int main()
{
    char s1[20], s2[20];
    float d;
    double e;
    scanf("%s", s1);
    scanf("%[^\n]", s2);
    scanf("%f %*d %lf", &d,&e);

    printf("\ns1=%s",s1);
    printf("\ns2=%s",s2);
    printf("\nd=%f  e=%lf",d,e);

    return 0;
}
```

6. Presmeravanje standardnog ulaza i izlaza

freopen("input.txt","r",stdin);
preusmerava standardni ulazni tok (stdin) na fajl input.txt radi citanja (r).

freopen("output.txt","w",stdout);
preusmerava standardni izlazni tok (stdout) na fajl output.txt radi upisa (w).

```
#include <stdio.h>
 void main() {
    int i, j;
    freopen("input.txt","r",stdin);
    freopen("output.txt","w",stdout);
     scanf("%d%d",&i,&j);
     printf("%d",i+j);
 }
```

4. NCP koji će ispisati na standardni izlaz ukupan broj prelazaka u novi red u tekstualnoj datoteci ulaz.txt. Pretpostaviti da je za ukupan broj prelazaka u novi red dovoljan opseg long promenljivih.

```
#include <stdio.h>
main()
{
   int znak; /*prihvata znak sa ulaza */
   long linije=0 ; /*brojac linija */
  freopen("ulaz.txt","r",stdin);
```

```
while ( (znak=getchar() ) != EOF)
  if (znak=='\n') linije ++;

printf("Prelazaka u novi red: %ld\n",linije);
}
```

7. NCP koji će tekst ulazne datoteke ulaz.txt iz foldera Z:\datoteke prepisati u izlaznu datoteku izlaz.txt u istom folderu tako da se višestruke uzastopne pojave znaka * zamene jednom zvezdicom.

ulaz (stdin, getchar())	**izlaz (stdout, putchar())**
1*3[tab]** a***d efg*+* [enter] ^Z	1*3[tab]* a*d efg*+* EOF

```
#include <stdio.h>
#define GRANICA '0'
main()
{
int znak; /*tekuci znak sa ulaza*/
int preth; /*znak koji prethodi tekucem */
preth=GRANICA;
freopen("Z:\\datoteke\\ulaz.txt","r",stdin);
freopen("Z:\\datoteke\\izlaz.txt","w",stdout);

while ( (znak=getchar() ) !=EOF)
 {
   if (znak !='*' || preth != '*') putchar(znak);
    preth=znak;
  }
}
```

8. Brojanje malih i velikih slova – uslovni izraz

Napisati C program koji čita tekst sa ulaza, prevodi sva mala slova u velika, i pri tom broji mala i velika slova. Program ilustruje operatore ',' (zarez), i ?:(uslovni operator).

```
#include <stdio.h>
int main()
{
int c, br_malih,  br_velikih;

/* Prepisujemo ulaz na izlaz, uz promenu malih slova u velika.
Takodje, brojimo mala i velika slova. */
for (br_malih = 0, br_velikih = 0 ; ( c = getchar() ) != EOF ; ) {
putchar(c >= 'a' && c <= 'z' ? c - 'a' + 'A' : c);
if (c >= 'a' && c <= 'z') br_malih++;
if (c >= 'A' && c <= 'Z') br_velikih++;
}

printf("Broj velikih slova: %d\nBroj malih slova: %d\n", br_velikih, br_malih);
return 0;
}
```

9. **(naredba break)** Napisati C program koji proverava korektnu uparenost zagrada (i) u tekstu koji dolazi sa standardnog ulaza. Npr. zagrade su korektno uparene u izrazu 5 *(8+3), a nisu u izrazu 3 +)2) ili izrazu ((3+6) * 8

```
#include <stdio.h>
main()
{
   int c;    /*tekuci karakter sa ulaza*/
   int br_otv = 0; /*brojac zagrada*/   /*citanje teksta karakter po karakter do kraja ulaza (EOF)  */
   while((c=getchar()) != EOF)
{
       if(c=='(')   br_otv++;
       if (c== ')')
          { br_otv--;
           if (br_otv<0)
                { printf("Visak zatvorenih zagrada\n"); break; }
          }
   }

    if (br_otv == 0) printf("Zagrade su u redu\n");
    else if (br_otv >0) printf("Visak otvorenih zagrada\n");
}
```

Naredba break se često koristi da se napusti beskonačna petlja. Pogledajmo petlju nalik na petlju iz prethodnog zadatka:

```
   while(1)
   {   c=getchar();
     if (c==EOF) break;
      if(c=='(')   br_otv++;
       if (c== ')')
          { br_otv--;
           if (br_otv<0)
                  { printf("Visak zatvorenih zagrada\n"); break; }
          }
   }
```

10. **(naredba continue)** NCP koji prihvata sa standardnog ulaza pozitivan ceo broj n (n <=50), proverava korektnost unete vrednosti, a zatim učitava n celih brojeva. Ispisati sumu nenegativnih brojeva na standardni izlaz.

```
#include <stdio.h>
main()
{
int n; /* dimenzija */
int indeks; /* brojac */
int suma, broj; /*suma nenegativnih clanova, ucitani
broj*//*unos dimenzije n i provera da li je taj broj u
segmentu 1..50 */

do
{ printf("Unesite broj elemenata niza\n");
scanf("%d", &n);
} while (n <1 || n > 50);

/* unos n celih brojeva i racunanje sume nenegativnih*/
for( indeks=0, suma=0; indeks<n; indeks++)
{
scanf("%d", &broj);
```

```
/* ignorisanje pozitivnih brojeva */
if(broj<0) continue;
suma+=broj; /*dodavanje nenegativnog broja sumi*/
}

printf("Suma nenegativnih je %d\n",suma);
}
```

11. NCP koji sadrzaj datoteke ulaz.txt sifrira tako sto se svaki znak koji je slovo ili cifra zamenjuje znakom ciji ASCII je veci za 1.

ULAZ	IZLAZ
BabaDeda	CbcbEfeb
01234 567	12345 678

12. NCP koji čita tekst sa standardnog ulaza i svaku linju teksta ispisuje na standardni izlaz šifrirane po shemi koja utiče samo na slova.

A B ... Y Z a b ... y z

c d ... a b D E ... B C

Broj linija teksta nije unapred poznat, linije su limitirane dužine. Može se pretpostaviti da mašinski set znakova odgovara ASCII kôdu.

PRIMER

ULAZ	IZLAZ
1. Baba 23/05	1. dDED 23/05
32. Zaza 34/04	32. bDCD 34/04

13. Za dato n odredi sumu
s= 1/1! - (1+2)/2! + (1+2+3)/3!- .... (1+2+...+n)/n!.

ULAZ	IZLAZ
n=1	1
n=2	-0.5
n=3	0.5
n=4	0.083333
n=5	0.208333
n=8	0.183829
n=9	0.183953
n=10	0.183938
n=12	0.183940

14. Za dato n i x odredi sumu:

$$s = \frac{x}{1} + \frac{x^2}{1+2} + \frac{x^3}{1+2+3} + \ldots + \frac{x^n}{1+2+\ldots n)}$$

ULAZ	IZLAZ
x=1 n=1	1
x=-1 n=3	-0.83333

x=2 n=2	3.333333
x=-2 n=10	10.294373
x=5 n=5	305
x=1 n=10	1.818182
x=8 n=8	556123.937500

15.Za dato n odredi sumu
s=1*2 + 2*3*4 + 3*4*5*6 + 4*5*6*7*8+ ... + n*(n+1)*...(2*n).

ULAZ	IZLAZ
n=1	2
n=2	26
n=3	386
n=4	7106
n=5	158306
n=6	4149986
n=7	125230946

16. Medju n realnih brojeva koji se ucitavaju sa tastature odrediti broj i njegov redni broj koji je najblizi nekom celom broju koji se zadaje sa standardnog ulaza.
primer: 4
brojevi: 3.3 4.98 5.01 8.89
resenje: broj: 5.01 redni broj:3

17.Napisati program kojim se unosi ispravan datum d, m, g i odredjuje redni broj tog dana u godini i koliko dana ima do kraja godine. (Godina je prestupna ako je deljiva sa 4 a nije sa 100, ili ako je deljiva sa 400).

ULAZ	IZLAZ	DO KRAJA GODINE
31.12.1800.	365	0
31.12.2000.	366	0
1.02.2009.	32	333
1.03.2009.	60	305
1.03.2008.	61	305
1.03.1900.	60	305

18.(a) Napisati program kojim se za dati prirodan broj n proverava da li je savrsen.
Definicija: Broj je savrsen ako je jednak sumi svojih delitelja iskljucujuci njega samog.
Npr.: 6 = 1+2+3
28 = 1+2+4+7+14
b) Napisati program kojim se ispisuju svi savrseni brojevi iz datog intervala [a,b].

19. Za dat prirodan broj n odredi najveci broj ciji je kub manji od n.Na primer: za n=45 rezultat je 3.

20. Za zadato Eps>0 naci najmanje N tako da je 2N/N!<Eps.Ispisati sve clanove niza od 1 do N.

21. Ispisati tablicu vrednosti funkcije X*X*X pri promeni X od 2 do N sa korakom 2.

22. Za dati prirodan broj N:
a) dodati 1 poslednjoj cifri broja. Primer: 37 ->38, 49 -> 410;
b) dodati 2 prvoj cifri broja. Primer: 49 ->69, 92-> 112;
c) udvojiti prvu cifru. Primer: 49->89, 89->169;
d) dodati 1 prvoj i poslednjoj cifri broja. Primer: 489->5810.

23. Brojevi se ucitavaju dok se ne unesu dva susedna jednaka broja. Odrediti koliko je brojeva u nizu. Na primer, ako se ucitavaju:
3 5 24 4 3 5 3 5 3 5 5
rezultat je: 11

24. Izracunati sa zadatom tacnoscu Eps sledece sume:
a) s=1 - x^2/2! + x^4/4! - ...
b) s=x - x^2/2 + x^3/3 - x^4/4 + ... za -1 < x <= 1
Sumirati do prvog clana koji je po apsolutnoj vrednosti manji od datog Eps.

25. **(okružno 2006, Srbija)**
Kompanija za izradu igračaka u kojoj ste zaposleni je počela da pravi novu vrstu robota. On ume da peva, govori, plače, rešava domaće zadatke i još dosta zanimljivih stvari. Svakako, jedna od najbitnijih karakteristika ovog robota je kretanje. Robot može da se kreće unapred ili da se okrene oko svoje ose za 90°. Zbog bolje orijentacije u prostoru, robotu je neophodno da uvek zna svoje koordinate. Kao članu projektnog tima koji se bavi softverom, dobili ste zadatak da na osnovu kretanja i okretanja robota odredite njegove koordinate.
Ulaz:
U prvom redu ulazne datoteke ZAD1.DAT, dati su prirodni brojevi N, X i Y, tako da važi $1 \leq N, X, Y \leq 1000$. N je broj komandi koje izvršava robot, a X i Y su početne koordinate robota u metrima. U drugom redu se nalazi N slova, od kojih je svaki jedno od tri slova: "D", "L" ili "N" (velikim slovima). "D" označava okretanje za 90° u smeru kazaljki na satu, "L" u smeru suprotnom od smera kazaljki, dok "N" predstavlja kretanje unapred. Dužina jednog robotovog koraka je 1 metar. Robot je u početku okrenut tako da gleda u pozitivnom smeru x-ose.
Izlaz:
U prvom i jedinom redu izlazne datoteke ZAD1.RES treba ispisati dva cela broja, X i Y, razdvojena razmakom, koji predstavljaju konačnu poziciju robota.
Primer:

ZAD1.DAT	ZAD1.RES
6 2 2 NNNLNN	5

26. **(Srbija, okružno 2007)** Na poljani se nalazi n zečeva i n zečica. Svi oni stoje duž jedne linije tako da svi zečevi gledaju desno niz liniju, a sve zečice levo niz liniju. Oni bi želeli da se podele u parove tako da u svakom paru bude jedna zečica i jedan zec koji mogu međusobno da se vide. Zec vidi zečicu ako je ona bilo gde desno od njega, a zečica vidi zeca ako je on bilo gde levo od nje. Od vas se traži da izbrojte na koliko načina oni to mogu uraditi tako da se svaki zec odnosno zečica nađe u tačno jednom paru. Pošto taj broj može biti veoma velik, nađite samo koliki ostatak daje taj broj pri deljenju sa 10007.

Ulaz:
(Ulazni podaci se nalaze u datoteci zecovi.in) U prvom redu zapisan je broj n. U drugom redu zapisano je tačno 2n simbola (bez razmaka) od kojih ima n znakova > koji označavaju zečeve (gledaju desno) i n znakova < koji označavaju zečice (gledaju levo).
Izlaz:
(Izlazne podatke upisati u datoteku zecovi.out) U prvom redu ispisati samo jedan broj - ostatak pri deljenju sa 10007 broja mogućih načina da se zečevi podele u parove.
Ograničenja:

- $1 \leq n \leq 100000$
- vremensko ograničenje za izvršavanje programa je 1 s.

Primer 1:

zecovi.in	zecovi.out
3 >><>><	4

Objašnjenje:
Ako zečeve i zečice označimo brojem koji odgovara poziciji na kojoj se nalaze, četiri moguća načina formiranja parova su:
{(1, 6), (2, 3), (4, 5)},
{(1, 3), (2, 5), (4, 6)},
{(1, 3), (2, 6), (4, 5)} i
{(1, 5), (2, 3), (4, 6)}

Primer 2:

zecovi.in	zecovi.out
8 >>>>>>>>><<<<<<	292

Objašnjenje:
Ukupan broj načina je 40320, što pri deljenju sa 10007 daje ostatak 292

27. **(Srbija, okružno 2008)** U popularnom televizijskom kvizu "Najslabija karika" nekoliko takmičara redom u krug odgovaraju na postavljena pitanja. Prvi tačan odgovor donosi jedan dinar, dok svaki sledeći tačan odgovor u nizu donosi duplo više poena nego prethodni. Kada neko odgovori netačno - lanac se prekida i sledeći tačan odgovor donosi 1 dinar. Kada takmičar kaže "Banka", sav novac u trenutnom lancu tačnih odgovora se dodaje na ukupnu zagarantovanu sumu, a naredni tačan odgovor ponovo vredi 1 dinar. Ako takmičari u nizu daju 15 tačnih odgovora, a između nije bilo bankiranja - nagrada iznosi neverovatnih 1000000000 (milijardu) dinara, bez obzira da li je neko rekao "Banka" ili su pre toga takmičari osvojili nešto novca. Vama su na raspolaganju podaci iz prethodne emisije. Treba odrediti koliko dinara su takmičari zaradili.
Ulaz:
(Ulazni podaci se nalaze u datoteci karika.in) U prvom redu ulazne datoteke se nalazi niz znakova. Svaki od znakova je iz skupa {'0', 'X', 'B'}, (sva tri moguća znaka su velika slova, što znaci da prvi od tri znaka nije cifra nula, vec slovo O) i oni predstavljaju tačan odgovor, netačan odgovor i banku. Dužina niza znakova je manja ili jednaka od 100000.
Izlaz:
(Izlazne podatke upisati u datoteku karika.out) U izlaznu datoteku treba upisati broj dinara koji su takmičari zaradili.
Primer 1:

karika.in	karika.out
OOOBXOXBXOOBOX	10

Objašnjenje.
Posle bankiranja, takmičari redom dobijaju 7 + 0 + 3 = 10 dinara.
Primer 2:

karika.in	karika.out
XOOBOOOOOOOOOOOOOOOXOOB	1000000000

Objašnjenje.
Takmičari imaju niz od 15 tačnih odgovora, pa zarađuju 1000000000 dinara.

28. **(Srbija 2008, okružno)** Na papiru je napisan broj N ($1 \leq N \leq 10^7$). Dva igrača naizmenično rade sledeće: igrač koji je na potezu bira neki delilac broja koji je napisan na papiru (delilac ne sme biti 1 ili broj sa papira), računa količnik broja sa papira i izabranog delioca, briše broj koji se nalazi na papiru, i piše dobijeni količnik. Ukoliko igrač ne može da odigra svoj potez (tj. jedini delioci su 1 i broj sa papira), on je pobednik. Napisati program koji treba da učita niz brojeva i za svaki od tih brojeva odredi koji igrač pobeđuje u partiji na čijem početku se na papiru nalazi taj broj. Smatrati da oba igrača igraju optimalno (povlače najbolje poteze).
Ulaz:
(Ulazni podaci se nalaze u datoteci igra.in) U prvom redu ulazne datoteke se nalazi broj M ($5 \leq M \leq 20$). U narednih M redova se nalazi po jedan broj Ni ($1 \leq Ni \leq 10^7$).
Izlaz:
(Izlazne podatke upisati u datoteku igra.out) U izlaznu datoteku treba upisati M redova. U i-tom redu ispisati broj 1, ako prvi igrač pobeđuje kad je početni broj na papiru Ni, a 2, ako drugi igrač pobeđuje.
Primer 1:

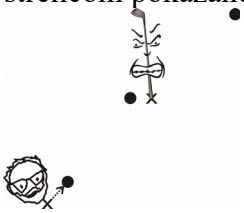
igra.in	igra.out
5 1 4 5 10 25	1 2 1 2 2

29. Đurajger Paunuds je profesionalni igrač golfa. Baš se danas vratio s turnira na kom je izvojevao veoma tešku, ali utoliko slađu pobedu, i sada zbog postturnirskog stresa ima problema s ružnim snovima. Đurajger sanja da se nalazi na ogromnom terenu za golf, na kom ga, sa željom da ga što dalje lansira, juri ogromna palica za golf! Srećom, na

ovom ogromnom terenu za golf ima mnogo ogromnih rupa (kao što je očekivano), i Đurajger je svoju šansu video u tome da se sakrije u jednu od njih. On treba da odabere jednu rupu i potrči k njoj pravolinijski. Palica odmah zaključuje ka kojoj rupi Đurajger trči, i potrčaće ka istoj rupi dvaput većom brzinom. Treba pronaći rupu ka kojoj Đurajger treba da potrči kako bi se uspešno spasao, ako takva postoji.
Ulaz:
(Ulazni podaci se nalaze u datoteci golf.in) U prvom redu ulazne datoteke nalaze se dva realna broja, koji predstavljaju x-koordinatu i y-koordinatu Đurajgerove početne pozicije. U drugom redu ulazne datoteke nalaze se još dva realna broja, koji predstavljaju x-koordinatu i y-koordinatu početne pozicije palice. U trećem redu nalazi se prirodan broj n ($1 \leq n \leq 1.000.000$), koji predstav a broj rupa na igralixtu. U narednih n redova nalaze se po dva realna broja, pri qemu se u i-tom od tih redova nalaze koordinate i-te rupe.
Izlaz:
(Izlazne podatke upisati u datoteku golf.out) U prvi i jedini red izlazne datoteke upisati redni broj rupe u koju Đurajger treba da se sakrije (ukoliko postoji više takvih, naći bilo koju), odnosno "Nadrljao je!" ukoliko takva rupa ne postoji.
Primer:

golf.in	golf.out
0.0 0.0 5.0 5.0 3 1.0 1.0 9.0 9.0 4.0 5.0	1

Objašnjenje.
Slika ispod prikazuje raspored rupa, kao i pozicije na kojima se nalaze Đurajger i palica za prvi primer. Takođe je strelicom pokazano koja je to rupa spasonosna za Đurajgera.



Primer:

golf.in	golf.out
0.0 0.0 5.0 5.0 2 9.0 9.0 4.0 5.0	Nadrljao je!

30. **(Srbija 2011, predokružno)** Data su tri broja A, B i C. Ispisati vrednost izraza A - B * C.
Ulaz.
(Ulazni podaci se učitavaju sa standardnog ulaza.) U prvom redu standardnog ulaza nalaze se prirodni brojevi A, B i C **($0 \leq A, B, C \leq 2^{32} - 1$)**.
Izlaz.
(Izlazne podatke ispisati na standardan izlaz.) U prvi i jedini red ispisati vrednost izraza A - B * C.
Ograničenja.
U 50% test primera će A, B i C imati vrednost iz intervala [0, 1000].
Primer 1.

standardni ulaz	standardni izlaz
1 2 3	-5

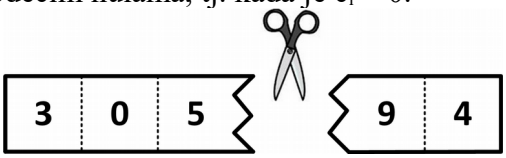
Objašnjenje.
1 - 2 * 3 = 1 - 6 = -5.
Primer 2.

standardni ulaz	standardni izlaz
0 0 0	0

Primer 3.

standardni ulaz	standardni izlaz
98574892 9473 9387590	-88830065178

31. **(Srbija 2012, okružno)** Data su dva prirodna broja n i m. Broj n možemo zapisati u dekadnom zapisu preko niza cifara kao $n = c_k c_{k-1} \ldots c_2 c_1$.
Nad brojem n je moguće primeniti operaciju sečenja između neke dve uzastopne cifre. Ako broj n isečemo na dva dela između cifara i + 1 i i, $i \in [1, k - 1]$ dobijamo dva nova broja $n1 = c_k \ldots c_{i+1}$ i $n2 = c_i \ldots c_1$. Pri ovom sečenju dozvoljen je i slučaj kada n2 počinje vodećim nulama, tj. kada je $c_i = 0$.



Primer sečenja broja n = 30594 pri čemu se dobija n1 = 305 i n2 = 94.

Za date brojeve n i m naći sečenje broja n takvo da je apsolutna razlika sume dobijenih delova, n1 +n2 , i broja m minimalna. Drugim rečima, naći sečenje koje minimizira izraz

$$| m - (n 1 + n 2) |$$

Ulaz.
(Ulazni podaci se nalaze u datoteci seksek.in.) U prvom i jedinom redu ulazne datoteke nalaze se dva prirodna broja n i m ($10 \leq n , m \leq 10^{18}$) opisana u tekstu problema. Brojevi ni m nemaju vodećih nula.
Izlaz.
(Izlazne podatke upisati u datoteku seksek.out.) U prvom i jedinom redu izlazne datoteke ispisati traženu vrednost (minimalnu apsolutnu razliku broja m i sume delova nekog sečenja broja n).
Primer 1.

seksek.in	seksek.out
30594 400	1

Objašnjenje.
Sva moguća sečenja i odgovarajuće apsolutne razlike za dati primer su:

```
n1 = 3 i n2 = 0594  197
n1 = 30 i n2 = 594  224
n1 = 305 i n2 = 94  1
n1 = 3059 i n2 = 4  2663
```