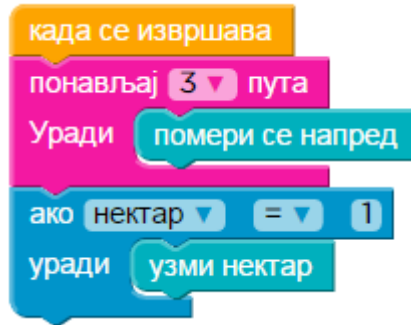


Uvod u programiranje - naredbe grananja



Uvežbavamo naredbe grananja

<https://studio.code.org/s/course2/stage/13/puzzle/3>



1. Programer Vasa želi da pomogne svojoj maloj sestri da provežba sabiranje i množenje jednocifrenih brojeva. Svaki put kada Vasa napiše jedan trocifren broj, njegova sestra mora da između svake dve uzastopne cifre postavi znak za sabiranje ili znak za množenje i izračuna dobijeni izraz. Na kraju mora da saopšti Vasi najveći rezultat koji je dobila. Vaš zadatak je da napišete program i pomognete Vasi da brzo i tačno pregleda sestrine zadatke. Vaš program treba da učitava jedan trocifren broj i ispiše najveću vrednost koja se dobije ako se između svake dve uzastopne cifre broja postavi znak za sabiranje ili znak za množenje.

PRIMER

ULAZ
128

IZLAZ
17

TEST PRIMERI ZA OCENJIVANJE

ULAZ	IZLAZ
111	3
109	10
999	729
400	4
199	82
913	27
990	81

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

```

int main()
{
    char a,b,c;
    cin >> a >> b >> c;
    int x=a-'0', y=b-'0', z=c-'0';
    int v = x+y+z;
    if(x+y*z>v) v=x+y*z;
    if(x*y+z>v) v=x*y+z;
    if(x*y*z>v) v=x*y*z;
    cout << v << endl;
}

```

2. Sa standardnog ulaza se unosi pozitivan četvorocifren ceo broj. Napisati program koji ispisuje njegovu najveću cifru na standardni izlaz.

ULAZ	IZLAZ
1111	1
1283	8
9123	9

3. Napisati program koji za 3 cela broja uneta sa standardnog ulaza ispisuje njihov maksimum na standardni izlaz.



PRIMER

ULAZ	IZLAZ
3 4 5	5
3 8 8	8
9 7 1000	1000
-100 -1 -100	-1



Hajde da ne pišemo odmah program već da sami osmislimo što bolje test primere.

Rešenje 01 – metod lažne pretpostavke

```

#include <stdio.h>
main(){

    int a, b, c;
    int max;
    printf("Unesite tri broja: ");
    scanf("%d%d%d", &a, &b, &c);
    max=a;
    if (max<b) max=b;
    if (max<c) max=c;
    printf("Makimum brojeva je: %d\n", max);

}

```

Zašto sledeća rešenja nisu korektna

Rešenje 02

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

    scanf ("%d%d%d", &a, &b, &c) ;

    if (a<b)
        if (a<c)          max=c;
        else              max=b;
    else
        if (b<c)
            max=c;
        else
            max=a;

    printf ("%d\n",max) ;
}
```

Uslovni izraz ? :

Slicno kao if.

izraz1 ? izraz2 : izraz3

$z = (a < b) ? a : b$; /*z=min(a,b); */

/*MOGLI SMO PISATI if (a<b) z=a; else z=b; */

VIŠESTRUKI if-else

if-else-else-else-...

if (izraz1) iskaz1

else if (izraz2) iskaz2

else if (izraz3) iskaz3

else iskaz4 /*ovaj if -else-else-else je jedna naredba !!!*/

if-if-...-else

VAŽNA NAPOMENA: else se odnosi na prvi neuparen if. Ako se želi drugačije moraju se navesti vitičaste zagrade.

if (izraz)

if (izraz1) iskaz 1

else iskaz

ovo else se odnosi na drugo if a ne na prvo if!!!

if-if-...-else

```

if (izraz)
{
    if (izraz1) iskaz 1
}
else iskaz

```

tek sada se else odnosi na prvo if!!!

Zadaci za samostalan rad

4. Napisati program koji izračunava i ispisuje zbir 4 najveća od pet brojeva unetih sa standardnog ulaza.

```

ulaz      izlaz
3 2 1 4 5  14

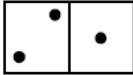
```

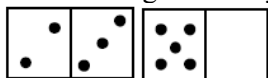
5. Sa standardnog ulaza se unose tri prirodna trocifrena broja. Spojiti dva najveća u šestocifren broj. Spajanje izvršiti tako da najveći od trocifrenih brojeva bude na početku šestocifrenog broja. Dobijeni šestocifreni broj ispisati na izlazu.

```

ULAZ      IZLAZ
185 247 311  311247

```

6. Miša se igra sastavljanja brojeva od domina. Na primer, dominom  je predstavljen broj 21, dok dominom



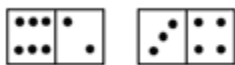
je predstavljen broj 2350. Napisati program DOMINE koji će ispisati na standardni izlaz što veći broj koji se može dobiti ređanjem dve domine čije vrednosti tačkica se učitavaju sa standardnog ulaza. Vrednosti se učitavaju u jednom redu i razdvojene su blanko karakterom. Na standardnom ulazu se najpre zadaju vrednosti tačkica sa prve domine, a potom sa druge domine. Smatrati da domine za igranje su pločice podeljene na dva polja, koja mogu biti oba prazna ili označena sa jednom do šest tačkica

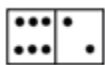
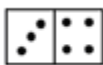
PRIMER

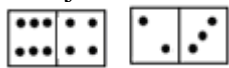
```

ULAZ          IZLAZ
2 6 3 4        6243

```



Objašnjenje: Domine sa ulaza su  , dok broj koji se očekuje na izlazu je 6243 koji se dobija ređanjem domine 6 2, a potom domine 4 3. Nikako se ne može dobiti broj 6432 kao rešenje, jer nam nisu date domine



. Napomena: priznaju se i rešenja u kom nije formiran višecifren broj, već samo ispisane njegove cifre.

ULAZ	IZLAZ
3 5 2 4	5342
5 1 4 5	5451
5 3 1 6	6153
4 0 5 0	5040
3 3 3 3	3333
6 4 6 2	6462
0 3 0 0	3000

```

#include <stdio.h>
int main()
{
    int dom11,dom12,dom21,dom22;
    int max1, max2, min1, min2;

    scanf("%d%d%d%d", &dom11,&dom12, &dom21, &dom22);

    //min i max tackice za 1. dominu
    max1=(dom11>dom12)?dom11:dom12;
    min1=(dom11<dom12)?dom11:dom12;

    //min i max tackice za 2. dominu
    max2=(dom21>dom22)?dom21:dom22;
    min2=(dom21<dom22)?dom21:dom22;

    if ((max1>max2) || ((max1==max2) && min1>=min2)) printf("%d%d%d%d\n",max1,min1,max2,min2);
    //printf("%d\n",1000*max1+100*min1+10*max2+min2);
    else if ((max1<max2) || ((max1==max2) && min1<min2)) printf("%d%d%d%d\n",max2,min2,max1,min1);
    //printf("%d\n",1000*max2+100*min2+10*max1+min1);
    return 0;
}

```

Makro zamene, cast-ovanje, operator sizeof

7. Korišćenjem makrozamene za protivrednost jednog evra u dinarima, NCP koji ispisuje na standardni izlaz dinarsku vrednost za četiri evra. Kako bi ispisali dinarsku protivvrednost jednog evra sa 0 decimala?

```

#include <stdio.h>
#define EURO 123.0477
main()
{
    printf("4 eura ima vrednost %f dinara\n", 4*EURO);
    /*printf("1 euro ima vrednost %.0f dinara\n",EURO);*/
}

```

8. NCP koji zadaje vrednosti dve realne promenljive i korišćenjem makrozamene naći kvadrat prve i maksimum njihovih vrednosti. Rezultat ispisati na standardni izlaz.

```

#include <stdio.h>
#define KVADRAT(x) ( (x)*(x) )
#define MAX(x,y) ( (x)>(y) ? (x) : (y) )
main()
{
    float w=3.2,y=8.8 ;
    printf("KVADRAT(%.2f)=%.2f\n",w,KVADRAT( w));
    printf("MAX(%.2f , %.2f)=%.2f\n",w,y,MAX(w,y));
}

```

IZLAZ

```
KVADRAT(3.20)=10.24  
MAX(3.20, 8.80)=8.80
```

RAZMIŠLJAMO

Da li su neke zagrade u definiciji makroa KVADRAT višak? Kakav bi bio efekat da je pisalo
KVADRAT(x) (x*x) i poziv u okviru printf da je bio KVADRAT(w+1)

Konverzija tipova – implicitna i eksplicitna

```
int a=180;
```

```
float b=18.88;
```

```
int c=a+b;
```

Šta mislite koji je rezultat naredbe `printf(“%d”, c);`

Pri računanju $a+b$ najpre se a konveruje u float, te vrednost izraza $a+b$ je 198.879999 .

Potom se vrši konverzija `int c= 198.879999` , a odatle `c=198`

Prijavljuje se upozorenje kompajlera da je možda došlo gubitka informacija, jer se širi rezultat smešta u užu promenljivu.

`char < int < long < float < double`

Eksplicitna konverzija

(tip) (izraz), kao na primer `(int) (2.8*2)` //vrednost izraza je 5

(tip) izraz, kao na primer `(int)2.8 * 2` //vrednost izraza je 4, tj. $2*2$ zbog veceg prioriteta cast operatora (int)

9. NCP koji sa standardnog ulaza učitava realan broj i ispisuje na standardni izlaz prvu decimalu (prvu cifru iza decimalne tacke).

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
float x; /*x je broj koji se unosi sa stdin*/
```

```
int d; /*d je prva decimala broja x*/
```

```
printf("Unesite realan broj: ");
```

```
scanf("%f",&x);
```

```

d=(int)(x*10)%10;
printf("\nPrva decimala broja %f jeste %d \n",x, d);
}

```

Relacioni i logički operatori

Relacioni operatori:

> >= < <= isti prioritet
 == != niži prioritet

(2003<2006)

(9<=10)

a < 5 != 1 ISTO KAO (a < 5)!=1

Logički operatori:

! unarna negacija (najviši prioritet)

&& logičko i (viši prioritet od ili)

|| logičko ili izračunavaju se sleva na desno!

5 && 4 vrednost je tačno

10 || 0 vrednost je tačno

0 && 5 vrednost je 0

!1 vrednost je 0

!9 vrednost je 0

!0 vrednost je 1

!(2>3) je 1

a>b && b>c || b>d je isto sto i ((a>b) && (b>c)) || (b>d)

koja je vrednost ako je a=10, b=5, c=1, d=15?

10.NCP koji na standardni izlaz ispisuje broj bajtova koji zauzimaju određeni tipovi podataka.

```
#include <stdio.h>
```

```
main()
```

```

{ printf("char: %d bajt\n", sizeof(char));
  printf("short: %d bajta\n", sizeof(short));
  printf("int : %d bajta\n", sizeof(int));
  printf("long=%d bajta\n", sizeof(long));
  printf("unsigned=%d bajta\n", sizeof(unsigned));
  printf("float=%d bajta\n", sizeof(float));
  printf("double=%d bajta\n", sizeof(double));
}

```

11. NCP koji na standardni izlaz ispisuje vrednosti predefinisanih konstanti (zaglavlja: limits.h, float.h).

```
#include <stdio.h>
#include <limits.h>
#include <float.h>
main()
{
    printf("CHAR_BIT:\t%d\n", CHAR_BIT);
    printf("UCHAR_MAX:\t%u\n", UCHAR_MAX);

    printf("INT_MIN:\t%d\n", INT_MIN);
    printf("INT_MAX:\t%d\n", INT_MAX);
    printf("LONG_MAX:\t%ld\n", LONG_MAX);
    return 0;
}
```

	1	2	3	4	5	6	7	8	9	10
1			■			■			■	
2		■			■			■		
3	■			●			■			■
4			■			■			★	
5		■			■			■		
6	■			■			■			■
7			■			■			■	
8		■			■			■		

12. Robot je zadužen za postavljanje pločica i rasvete u kuhinji. Za kuhinju se koriste pločice u žutoj i narandžastoj boji i to tako što se između dve narandžaste pločice postavljaju dve žute pločice. Prvi red pločica započinje sa dve žute pločice, potom jedna narandžasta, potom opet dve žute, potom narandžasta i tako do kraja reda. Naredni red započinje jednom žutom, potom jedna narandžasta, zatim dve žute i tako do kraja reda. Treći red započinje jednom narandžastom pločicom, zatim dve žute, pa jedna narandžasta i takp dalje. Ova tri reda se ponavljaju dok se ne pokrije cela kuhinja. Za osvetljenje se koriste dve vrste lampi, jedna sa oznakom 1 i druga sa oznakom 2. Zbog posebnog načina postavljanja instalacija, osvetljenje radi samo ako se dve lampe iste vrste postave na pločice koje su iste boje. Pločice se u svakom redu, sleva-nadesno označe brojevima 1, 2, 3, i tako dalje, a svaki red „odpzo-nadole“ se, takođe označi brojevima 1, 2, 3, i tako dalje. Napisati program u kome se unosi vrsta i pozicija za obe lampe, a program ispisuje komentar o tome da li rasveta radi.

Ulazni podaci. Standardni ulaz sadrži šest linija od kojih svaka sadrži po jedan pozitivan ceo broj i to sledećim redim: prva linija – **L1** vrsta prve lampe (1 ili 2), druga linija – **X1** broj reda u koji se postavlja prva lampa, treća linija – **Y1** broj pločice u redu na koju se postavlja prva lampa, četvrta linija **L2** – vrsta druge lampe (1 ili 2), **X2** – broj reda u koji se postavlja druga lampa, **Y2** – broj pločice u redu na koju postavlja druga lampa ($0 < X1, Y1, X2, Y2 \leq 10\ 000$).

Izlazni podaci. Jedina linija standardnpg izlaza sadrži jedan od 4 komentara:

RADI – što označava da će rasveta raditi

LOSE LAMPE – što označava da su lampe različite vrste, ali na pločicama iste boje

LOSA POZICIJA – što označava da su lampe iste vrste, ali na pločicama različitih boja

POTPUNO NEPOKLAPANJE – što označava da se ne poklapaju ni vrste lampi ni boje pločica.

Primer

```

Ulaz:  Izlaz:
1      LOSE LAMPE
3
4
2
4
9

```

L1	X1	Y1	L2	X2	Y2	Излаз
1	3	4	2	4	9	LOSE LAMPE
1	4	6	1	12	7	RADI
2	15	25	2	12	24	LOSA POZICIJA
2	22	8	2	14	12	RADI
1	8	8	2	16	16	POTPUNO NEPOKLAPANJE
1	11	11	1	15	11	LOSA POZICIJA
2	10	16	1	8	19	LOSE LAMPE

13. Mali Marko mnogo voli kroasane. U njegovoj omiljenoj pekari prodaju se tri vrste kroasana – čokoladni, vanila, sa džemom. Marko je veoma gladan, i želi da kupi što je moguće više kroasana. Naravno, broj kroasana u pekari je ograničen, ali je ograničena i suma novca koju ima mali Marko. Moguće je i da neki kroasani koštaju 0 dinara.

Napišite program i pomozite Marku da nađe najveću moguću količinu kroasana, koju može da kupi.

Ulaz

U prvom redu standardnog ulaza se nalaze tri cela broja – cena kroasana sa čokoladom, vanilom, sa džemom. U drugom redu se nalaze tri cela broja – broj kroasana sa čokoladom, vanilom, džemom koji se mogu kupiti u pekari. Brojevi su razdvojeni s jednim blanko karakterom.

U trećem redu se nalazi jedan ceo broj – novac s kojim raspolaže Marko.

Izlaz

U jednom redu standardnog izlaza programa treba da se ispiše jedan ceo broj – najveći broj kroasana koji može da kupi Marko.

Ograničenja

Svi brojevi su celi nenegativni brojevi, ne veći od 100 000 000 000 000 000

Primer 1

Ulaz

5 3 8
2 6 4
23

Izlaz

7

Primer 2

Ulaz

15 18 20
1 4 100
1000000

Izlaz

105

Primer 3

Ulaz

47563546 177777 3524
0 4757 0
4758698

Izlaz

26

Primer 4

Ulaz

0 0 0
240000 90000 50000
9000000

Izlaz

380000