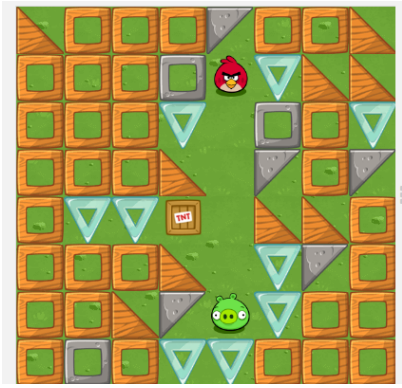
8. čas

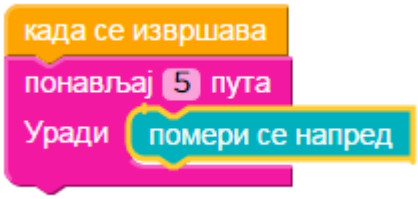
Uvod u programiranje - naredbe ciklusa



Uvežbavamo naredbe ciklusa,

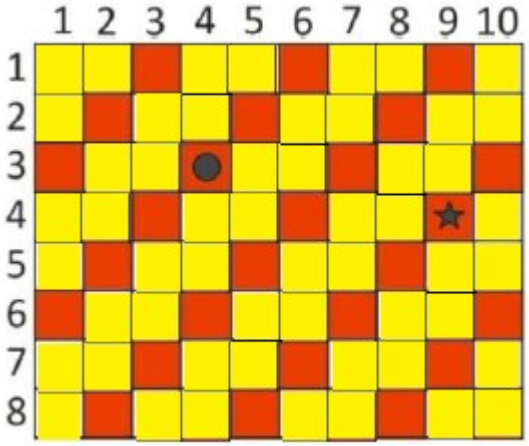
naredbe ponavljanja (loop)

https://studio.code.org/s/course2/stage/6/puzzle/3



"Talk is cheap. Show me the code."
- Linus Torvalds

Zadaci za domaći sa prethodnog časa



1. Robot je zadužen za postavljanje pločica i rasvete u kuhinji. Za kuhinju se koriste pločice u žutoj i narandžastoj boji. 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 tako 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 „odozgo-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 standardnog 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 3	LOSE LAMPE

	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

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

int main(){
  int l1,l2,x1,x2,y1,y2,m1,m2;
    cin >> l1 >> x1 >> y1;
    cin >> l2 >> x2 >> y2;

   m1=(x1+y1)%3;
   m2=(x2+y2)%3;

  if (l1==l2) {   /* ne smemo pisati  if (l1=l2)  */
     if (m1==m2) cout << "RADI\n";
     else if ((m1!=1) && (m2!=1)) cout << "RADI\n";
   else cout << "LOSA POZICIJA\n";
   }
  else {
     if (m1==m2) cout << "LOSE LAMPE\n";   /* ne smemo pisati  if (m1=m2)  */
    else if ((m1!=1) && (m2!=1)) cout << "LOSE LAMPE\n";
     else cout << "POTPUNO NEPOKLAPANJE\n";
  }
  return 0;
}
```

Operator zarez

Operator zarez (,) se koristi da razdvoji dva ili više izraza na mestu gde je samo jedan izraz očekivan.
Na primer:
a = (b=5, b+2);
Prvo će vrednost 5 biti dodeljena promenljivoj b, a onda će b+2 biti dodeljeno promenljivoj a.
Na kraju će promenljiva **a** sadržati vrednost 7, dok će promenljiva **b** sadržati vrednost 5.

U naredbi

```
 min4=( (m=(a<b)? a:b), (n=(c<d)?c:d), ((m<n) ? m:n));
```

1. najpre će se izvršiti naredbe dodele
m=(a<b)? a:b
n=(c<d)?c:d
nakon kojih m i n uzimaju redm minimalne vrednosti tako da m=minimum(a,b), n=minimum(c,d)

2. promenljivoj min4 će se dodeliti vrednost poslednjeg izraza u nizu tj.((m<n) ? m:n))
odnosno minimum(a,b,c,d)

Pored mogućnosti da skrati zapis pojedinih delova programa, upotreba ovog operatora krije i moguće greške.
Više o tome kad budemo radili nizove i matrice.

---

2. 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	**Primer 2**	**Primer 3**	**Primer 4**
Ulaz 5 3 8 2 6 4 23	**Ulaz** 15 18 20 1 4 100 1000000	**Ulaz** 47563546 177777 3524 0 4757 0 4758698	**Ulaz** 0 0 0 240000 90000 50000 9000000
Izlaz 7	**Izlaz** 105	**Izlaz** 26	**Izlaz** 380000

Rešenje

Da bi se maksimizirala količina kupljenih kroasana, logično je da se počne s najeftinijim kroasanom. U početku potrebno je da razmenimo mesta datim cenama c1, c2, c3 tako da budu sortirane (poređane) neopadajuće tj. c1<=c2<=c3. Dakle, c1 je najjeftinija cena, c3 je najskuplja cena. Razmenu ćemo izvršiti funkcijom trampe **swap** iz zaglavlja **algorithm**. Evo kako radi funkcija swap

```
#include <algorithm>    // std::swap  ili using namespace std;

int x=10, y=20;       // x:10 y:20
std::swap(x,y);        //x postaje 20, y postaje 10.
```

Kako bi uradili trampu vrednosti dve promenljive bez gotove funkcije swap?

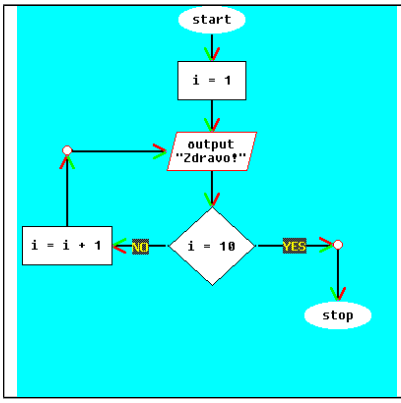
Slično kada god trampimo, na primer c1, c2, moramo da trampimo i količinu kroasana n1, n2.
Slično kada god trampimo, na primer c2, c3, moramo da trampimo i količinu kroasana n2, n3.

Označimo sa s - novac s kojim raspolaže Marko.

Dakle, Petar može da od prvog tipa (najjeftinijih) kroasana uzme najviše *s/c1*, ali taj količnik ne sme da bude veći od ukupne raspoložive količine tih kroasana, tj. vrednosti promenljive *n1*.
Zato koristimo formulu: min {n1, [s/c1]}.
Analogno od drugog tipa kroasana može najviše da uzme: min {n2, [s/c2]}.
Analogno od trećeg tipa kroasana može najviše da uzme: min {n3, [s/c3]}.
Pre svakog deljenja s/c1 ili s/c2 ili s/c3 proveriti da li su c1, c2, c3 različiti od 0 (jer nije dozvoljeno deliti sa 0).
Prema veličinama datim u formulaciji zadatka, moramo koristiti tip `long long`.

```
#include <iostream>
#include <cstdio>
#include <algorithm>
using namespace std;
int main()
{ long long c1, c2, c3, n1, n2, n3, s;
//c1,c2,c3 cena kroasana sa cokoladom, vanilom, dzemom
//n1,n2,n3 broj kroasana sa cokoladom, vanilom, dzemom
//s - novac s kojim raspolaze Marko
cin>>c1>>c2>>c3>>n1>>n2>>n3>>s;
if(c1>c2) swap(c1,c2), swap(n1,n2);
if(c2>c3) swap(c2,c3), swap(n2,n3);
if(c1>c2) swap(c1,c2), swap(n1,n2);
long long r, t, pom;
if(c1!=0) pom=s/c1; else pom=n1;
r = min(n1, pom);
s=s-r*c1;
if(c2!=0) pom=s/c2; else pom=n2;
t = min(n2, pom);
s= s-t*c2;
r = r+t;
if(c3!=0) pom=s/c3; else pom=n3;
r = r+min(n3, pom);
cout<<r<<endl;
return 0;
}
```

3.

	Napisati C/C++ program koji ispisuje 9 puta poruku ***Zdravo!*** Svaku poruku ispisati u posebnom redu

while (uslov) **telo petlje**	**do naredba** **while (izraz);**	**for (izraz1; izraz2; izraz3)** **naredbe** **/* isto kao */** **izraz1;** **while (izraz2)**

		{
		naredbe
		izraz3;
		}

#include <stdio.h>
int main(){
int x;
x = 1;
while (x<10){
printf("Zdravo!\n");
x++;
}
 	
return 0;
}
 	
/* x++ je isto kao i x=x+1 */
	#include <stdio.h>
	int main(){
	int x;
	x = 1;
	do{
	printf("Zdravo!\n");
	++x;
	} while (x<10);
	
	return 0;
	}
	
	/* ++x je isto kao i x=x+1 */
		#include <stdio.h>
		int main()
		{
		int x;
		for (x = 1; x < 10; x++)
		printf("Zdravo!\n");
		
		return 0;
		}

4. Napisati C/C++ program koji ispisuje brojeve od 1 do 9 u formatu
x = 1, x = 2, x = 3, x = 4, x = 5, x = 6, x = 7, x = 8, x = 9,

while (uslov) telo petlje	do naredba while (izraz);	for (izraz1; izraz2; izraz3) naredba

#include <iostream>
using namespace std;
int main(){
int x;
x = 1;
while (x<10){
cout << "x = " << x<<", ";
//printf("x = %d, ",x);
x++;
}
 	
return 0;
}
	#include <iostream>
	using namespace std;
	int main(){
	int x;
	x = 1;
	do{
	cout << "x = " << x<<", ";
	//printf("x = %d, ",x);
	++x;
	} while (x<10);
	
	return 0;
	}
		#include <iostream>
		using namespace std;
		int main()
		{
		int x;
		for (x = 1; x < 10; x++)
		cout << "x = " << x<<", ";
		//printf("x = %d, ",x);
		
		return 0;
		}

Izlaz iz programa će biti:
x = 1, x = 2, x = 3, x = 4, x = 5, x = 6, x = 7, x = 8, x = 9,

4. Napisati C/C++ program koji štampa parne brojeve od 1 do 10.

#include <stdio.h>
int main(){
int x;
x = 2;
while (x<=10){
	#include <stdio.h>
	int main(){
	int x;
	x = 2;
	do{
		#include <stdio.h>
		int main()
		{
		int x;
		for (x = 2; x <=10; x+=2)

printf("x = %d, ",x);
x=x+2;
}
 	
return 0;
}
	printf("x = %d, ",x);
	x+=2;
	} while (x<=10);
	
	return 0;
	}
		printf("x = %d, ",x);
		
		return 0;
		}

5. Šta je ispisuje sledeći program?

```
#include <stdio.h>
int main() {
int x , y, z;
y = 1;
while( y < 5 ) { x = y++; z = ++y; }
printf("%d %d %d\n", x, y, z);
return 0;
}
```

Pratimo vrednosti promenljivih
y=1
x=1 y=2 y=3 z=3
x=3 y=4 y=5 z=5

RESENJE:
3 5 5

6. Šta je ispisuje sledeći program?

```
#include <stdio.h>
int main() {
int x, y;
for(x = 0, y = 1000; y > 1; x++, y /= 10);
printf("%d %d\n", x, y);
return 0;
}
```

Pratimo vrednosti promenljivih
x=0 y=1000

x=1 y=100
x=2 y=10
x=3 y=1

RESENJE:
3 1

7. **INFINITE LOOP**
Beskonačna petlja se može realizovati pomoću while petlje:

```
 while(i=1) { printf("%d ", i);}
```

Program sa beskonačnom petljom

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

```
 main()
{ int i;
  while(i=1) { printf("%d ", i);}
}
```

Napišite C/C++ program sa beskonačnom petljom koji koristi for ciklus.

8. Za dati for ciklus napravite isti takav while ciklus.

`for(i=1;i<=n;i++){printf("%d",i);}`	`i=1;`
	`while(i<=n)`
	`{ printf("%d",i);`
	`  i++;`
	`}`

`for(i=1,j=10; (i<=n) && (j>=0);i++, j--)`
`{printf("%d ",i); printf("%d ",2*j);}`
	`i=1; j=10;`
	`while((i<=n) && (j>=0))`
	`{ printf("%d ",i);`
	`  printf("%d ",2*j);`
	`  i++; j--;`
	`}`

9. Napisati C/C++ program, koji učitava pozitivan ceo broj sa standardnog ulaza i ispisuje najveću cifru tog broja.

```
ulaz    izlaz
120387    8
95915     9
```

```
#include <stdio.h>
main()
{
short cifra=0, max=0; /*tekuca cifra broja, max cifra*/
unsigned a; /*ucitani broj*/
printf("Unesite broj: ");
scanf("%u", &a);

  while(a!=0)
    {cifra=a%10;
     if (cifra >max) max=cifra;
     a=a/10;
    }

printf("\nMax cifra je: %hd\n",max);
}
```

10. Napisati C/C++ program koji učitava 50 realnih brojeva sa standardnog ulaza, a ispisuje na standardni izlaz njihov zbir.

IDEJA: U ovom zadatku treba sabrati do 50 vrednosti, ali je nepraktično koristiti do 50 promenljivih za čuvanje unetih vrednosti, a zatim sabrati vrednosti tih 50 promenljivih. Zato se pribegava upotrebi ciklusa: tj. 50 puta će se vršiti učitavanje vrednosti sa tastature u promenljivu x, i nakon svakog učitavanja vrednost promenljive x će se dodati zbiru. Na početku je zbir jednak 0.

```
#include <stdio.h>
main ()
```

```
{
        int i=1; /*brojac ucitanih brojeva */
        float x, zbir; /* broj sa stdin, suma unetih brojeva */
        for(zbir=0, i=1; i<=50;i++) {scanf("%f",&x); zbir=zbir +x;}
        printf("\nZbir = %f\n", zbir);
}
```

Domaci zadaci (ne morate uraditi sve domace zadatke, vec koliko mozete)

1. Napisati progam koji za uneto n (n>0) ispisuje prvih n parnih brojeva.
Na primer: za ulaz 4 program ispisuje 2 4 6 8

2. Napisati program koji za uneto n (n>0) na izlazu ispisuje faktorijel broja n (proizvod prvih n brojeva).
Na primer: za ulaz 4 izlaz je 24, za ulaz 5 izlaz je 120, jer 1*2*3*4=24, 1*2*3*4*5=120

3. Napisati program koji za unete brojeve x i n (n>0) na izlazu ispisuje n-ti stepen broja x.
Na primer: za ulaz 2 3 izlaz je 8, za ulaz 4 2 izlaz je 16, jer 2*2*2=8, 4*4=16

4. Napisati program koji za uneti broj n (n>0) ispisuje zbir njegovih cifara.
Na primer: za ulaz 123 izlaz je 6, za ulaz 5403 izlaz je 12

5. Napisati program koji koristi for petlju. Unosi se broj elemenata, a zatim i sami elementi.
Program ispisuje najveci od unetih brojeva, kao i aritmeticku sredinu unetih brojeva.

```
ULAZ                IZLAZ
5                     Aritmeticka sredina:3.400000
1 2 3 4 7                Najveci broj:7.000000
```

6. Napisati C program koji učitava pozitivan ceo broj sa standardnog ulaza i ispisuje najmanju cifru tog broja.

```
ulaz    izlaz
84271    1
90492    0
```

7. Napisati C program koji učitava sa standardnog ulaza prirodan broj manji od milijarde i na standardni izlaz ispisuje da li cifre tog broja obrazuju strogo rastući niz.
Na primer, cifre broja 118 ne obrazuju strogo rastući niz,
cifre broja 8876551 ne obrazuju strogo rastući niz, dok cifre broja 1234569 obrazuju strogo rastući niz.

8. Napisati C program koji učitava sa standardnog ulaza prirodan broj manji od 100000 i na standardni izlaz ispisuje da li je taj broj Armstrongov. N-to cifren broj je Armstrongov ako je jednak sumi N-tih stepena svojih cifara. Na primer, 1002 nije, 370 jeste, 407 jeste.