

Zadaci 1.X	
Predmet:	Programiranje I
Autor:	Kemal Marić
Kontakt informacije:	kemal@edu.fit.ba ; ilhan@edu.fit.ba

Primjer #1	<p>Neka je data formula za računanje sume S:</p> $S = \frac{1}{100} \sum_{i=1}^{100} i$ <p>Potrebno je pretvoriti datu formulu u programski kod.</p>
Rješenje	<pre>void main(){ float S = 0.0f; for (int i = 1; i <= 100; i++) S += i; S /= 100; cout << "Suma je: " << S << endl; }</pre>
Ovo je primjer korištenja <i>for</i> petlje za računanje aritmetičke sredine brojeva, koji se nalaze u opsegu [1,100].	

Primjer #2	<p>Napisati programski kod za računanje proizvoda/produkta:</p> $P = \frac{1}{10} \prod_{i=1}^{10} i$
Rješenje	<pre>void main(){ float P = 1.0f; for (int i = 1; i <= 10; i++) P *= i; P /= 10; cout << "Produkt je: " << P << endl; }</pre>

Zadatak 1. X1

Napisati programski kod za računanje sume S na osnovu sljedeće formule:

$$S = \sum_{i=1}^n \frac{i^2}{\sqrt{i}}, 1 \leq n \leq 100,$$

Potrebno je unijeti cjelobrojnu vrijednost n , takvu da se nalazi u zadanom opsegu vrijednosti ($1 \leq n \leq 100$)(ponavljati unos u slučaju da je n izvan opsega). Zatim je potrebno uraditi sumiranje svih vrijednosti, te ispis konačne vrijednosti sume S .

Hint #1: Koristiti *do while* petlju za ponavljanje unosa n u slučaju da se n ne nalazi u zadanom opsegu.

Zadatak 1. X2

Napisati programski kod za računanje sume S na osnovu sljedeće formule:

$$S = \sum_{u=1}^5 u!$$

Nakon izvršenja koda za sumiranje vrijednosti, potrebno je ispisati vrijednost sume S .

Hint #1: $u!$ označava faktorijel broja u , koji se računa po formuli:

$$u! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot u$$

$$\text{e.g. } 10! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot \dots \cdot 10$$

Hint #2: Koristiti dvostruku petlju u ovom primjeru (vanjsku za računanje sume te unutarnju za računanje faktorijela).

Zadatak 1. X3

Napisati programski kod za računanje sume S na osnovu sljedeće formule:

$$S = \sum_{i=-5}^n \left| \frac{i}{n} \right|, n \neq 0, n \geq -5$$

Nakon izvršenja koda za sumiranje vrijednosti, potrebno je ispisati vrijednost sume S .

Hint #1: Za računanje absolutne vrijednosti koristiti matematičku funkciju *abs*

Hint #2: Uraditi konverziju varijable i u *float* da bi se osiguralo da je rezultat dijeljenja $\frac{i}{n}$ decimalni broj (a ne *int*). Za bolje razumijevanje posmatranog aspekta, pogledati razlike u vrijednostima rezultata izvršenja za sljedeće tri linije koda:

```
cout << 1 / 6 << endl; //0
cout << float(1)/ 6<< endl; //0.166667
cout << 1 / float(6) << endl; //0.166667
```

Zaključak je da ukoliko želimo izbjeći cjelobrojno dijeljenje, neophodno da je konvertujemo ili brojnik ili nazivnik u tip *float/double* ili oba.

Zadatak 1. X4

Napisati programski kod za računanje sume S na osnovu sljedeće formule:

$$S = \sum_{i=n}^{2n} 2^{|i-2|}, n \geq 0$$

Nakon izvršenja koda za sumiranje vrijednosti, potrebno je ispisati vrijednost sume S .

Zadatak 1. X5

Napisati programski kod za računanje sume S na osnovu sljedeće formule:

$$S = \frac{1}{\sqrt{5}} \sum_{i=-10}^{10} (i^2) - \frac{1}{\sqrt{5}} \sum_{i=-5}^5 (i)$$

Nakon izvršenja koda za sumiranje vrijednosti, potrebno je ispisati vrijednost sume S .

Zadatak 1. X6

Napisati programski kod za računanje sume S na osnovu sljedeće formule:

$$S = \sum_{i=1}^n \sum_{j=5}^{10} (i^2 j - j^2 i), n > 5$$

Nakon izvršenja koda za sumiranje vrijednosti, potrebno je ispisati vrijednost sume S .

Hint #1: Koristiti „ugniježdenu“ petlju budući da je u ovom primjeru prisutna dvostruka suma sa dva iteratora (varijable) i i j

Zadatak 1. X7

Napisati programski kod za računanje sume S na osnovu sljedeće formule:

$$S = \sum_{i=1}^n \left(i^2 + \sum_{j=5}^{10} \frac{(i-j)^2}{j^2} \right), 1 \leq n < 16$$

Nakon izvršenja koda za sumiranje vrijednosti, potrebno je ispisati vrijednost sume S .

Hint #1: Koristiti „ugniježdenu“ petlju budući da je u ovom primjeru prisutna dvostruka suma sa dva iteratora i i j

Zadatak 1.X8

Napisati programski kod za računanje produkta P na osnovu sljedeće formule:

$$P = \prod_{i=1}^n (i + 1)^{-1}, 1 \leq n \leq 10$$

Nakon izvršenja koda za množenje vrijednosti, potrebno je ispisati vrijednost produkta P .

Zadatak 1.X9

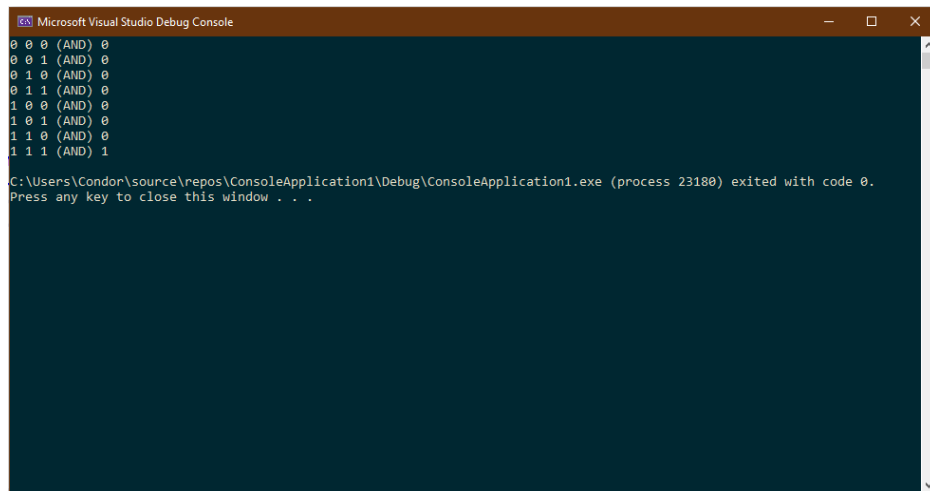
Napisati programski kod za računanje produkta P na osnovu sljedeće formule:

$$P = \prod_{i=1}^{10} \sum_{j=-5}^5 (i - 1) \cdot (1 - j)$$

Nakon izvršenja koda za množenje vrijednosti, potrebno je ispisati vrijednost produkta P .

Zadatak 1.X10

Napisati programski kod (korištenjem petlji) za računanje tabele istinitosti za AND kolo za tri logička ulaza u_1, u_2, u_3 . Izlaz iz programa treba biti u sljedećoj formi:



```
Microsoft Visual Studio Debug Console
0 0 0 (AND) 0
0 0 1 (AND) 0
0 1 0 (AND) 0
0 1 1 (AND) 0
1 0 0 (AND) 0
1 0 1 (AND) 0
1 1 0 (AND) 0
1 1 1 (AND) 1

C:\Users\Condor\source\repos\ConsoleApplication1\Debug\ConsoleApplication1.exe (process 23180) exited with code 0.
Press any key to close this window . . .
```

Hint #1: Koristiti trostruku (troslojnu) *for* petlju sa varijablama u_1, u_2 i u_3 .

Zadatak 1.X11

Napisati programski kod za računanje tabele istinitosti za *NOR* kolo (*Negative OR*) za tri logička ulaza: u_1, u_2 i u_3 . Izlaz iz programa treba biti u sljedećoj formi:

```
Microsoft Visual Studio Debug Console
0 0 0 (NOR) 1
0 0 1 (NOR) 0
0 1 0 (NOR) 0
0 1 1 (NOR) 0
1 0 0 (NOR) 0
1 0 1 (NOR) 0
1 1 0 (NOR) 0
1 1 1 (NOR) 0

C:\Users\Condor\source\repos\ConsoleApplication1\Debug\ConsoleApplication1.exe (process 13964) exited with code 0.
Press any key to close this window . . .
```

Hint #1: Koristiti trostruku (troslojnu) *for* petlju sa varijablama u_1, u_2 i u_3 .

Zadatak 1.X12

Napisati programski kod za računanje tabele istinitosti za složeno logičko kolo sa tri ulaza: u_1, u_2 i u_3 , koje je određeno sljedećom formulom:

$$\neg(u_1 \vee u_2) \wedge u_3$$

Simboli	
\neg	Negacija
\wedge	AND
\vee	OR

Izlaz iz programa treba biti u sljedećoj formi:

```
Microsoft Visual Studio Debug Console
0 0 0 (operacija) 0
0 0 1 (operacija) 1
0 1 0 (operacija) 0
0 1 1 (operacija) 0
1 0 0 (operacija) 0
1 0 1 (operacija) 0
1 1 0 (operacija) 0
1 1 1 (operacija) 0

C:\Users\Kemal\source\repos\TestingGround\Debug\TestingGround.exe (process 8648) exited with code 0.
Press any key to close this window . . .
```

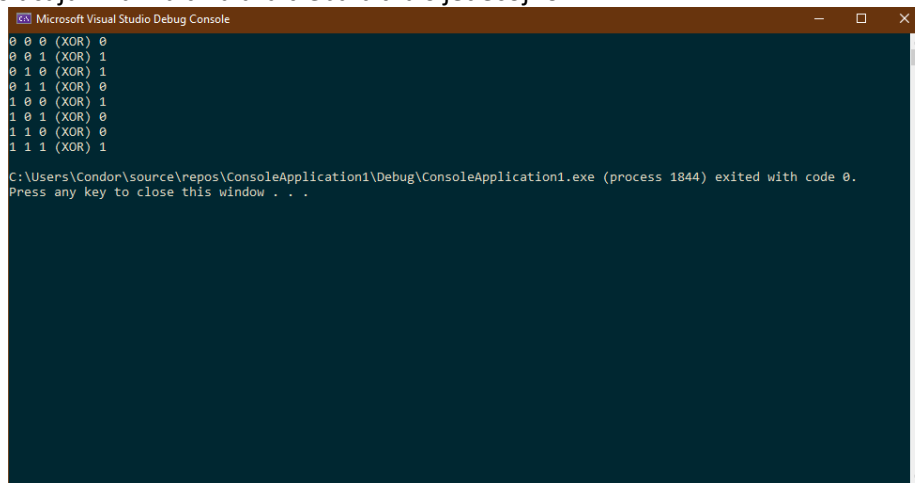
Hint #1: Koristiti trostruku (troslojnu) *for* petlju sa varijablama u_1, u_2 i u_3 .

Zadatak 1. X13

Napisati programski kod za računanje tabele istinitosti za *XOR* kolo (*Exclusive OR*) za tri logička ulaza u_1, u_2, u_3 . Primjer XOR kola za dva ulaza je dat na sljedećoj slici:

Ulaz		Izlaz
u_1	u_2	$u_1 XOR u_2$
0	0	0
0	1	1
1	0	1
1	1	0

Rezultat XOR kola za dva ulaza je logička vrijednost 0 (*false*) ukoliko su ulazi isti, te 1 (*true*) u suprotnom slučaju. Izlaz za tri ulaza treba biti u sljedećoj formi:



```
Microsoft Visual Studio Debug Console
0 0 0 (XOR) 0
0 0 1 (XOR) 1
0 1 0 (XOR) 1
0 1 1 (XOR) 0
1 0 0 (XOR) 1
1 0 1 (XOR) 0
1 1 0 (XOR) 0
1 1 1 (XOR) 1
C:\Users\Condor\source\repos\ConsoleApplication1\Debug\ConsoleApplication1.exe (process 1844) exited with code 0.
Press any key to close this window . . .
```

Hint #1: Koristiti trostruku (troslojnu) *for* petlju sa varijablama u_1, u_2 i u_3 .

Hint #2: Možete koristiti C++ *bitwise* operator '^', koji primjenjuje *XOR* logičku operaciju nad dva ulaza. O *bitwise* operatorima će biti više riječi u sklopu budućih vježbi.

Zadatak 1. X14

Napisati program koji će zahtijevati od korisnika unos suglasnika (ponavljati unos dokle god uneseno slovo nije suglasnik). Za uneseno slovo ispisati njemu susjedna slova, pri čemu ako je uneseno slovo 'Z', onda su njemu susjedna slova 'Y' i 'A'.

Hint #1: Koristiti ASCII šemu za konverziju između *int* i *char* tipova podataka

```
cout << int('K') << endl; // 75
cout << char(75) << endl; //'K'
```

Hint #2: Koristiti *do while* petlju, koja će se ponavljati dokle god korisnik unosi samoglasnike (a ne suglasnike).

Zadatak 1. X15

Napisati program koji će ispisati sva mala slova sa parnim indeksom. Uraditi prilagodbu ASCII šeme prilikom izrade ovog zadatka.

Indeks	Slovo
0	'a'
1	'b'
2	'c'
3	'd'
...	
25	'z'

Rezultat programa: a, c, e, ...

Zadatak 1. X16

Napisati program koji će omogućiti korisniku unos jednog slova (velikog ili malog) te prirodnog broja n . ($n > 0$). Zatim je potrebno uraditi pomak onoliko puta, kolika je vrijednost unesenog n . Ukoliko pomak prevazilazi granice engleskog alfabeta, potrebno je vratiti se na početak.

Slovo	Pomak	Rezultat
'a'	5	'f'
'b'	2	'd'
'z'	1	'a'
'A'	5	'F'
'Z'	2	'B'
'k'	1	'l'

Hint #1: U situacijama gdje pomak prevazilazi granice alfabeta možete koristiti aritmetički operator modulo '%' na način da se dobije tačna vrijednost pomaka u odnosu na početak alfabeta.

Zadatak 1. X17

Napisati program koji omogućuje korisniku unos prirodnog broja n (gdje je $10 \leq n \leq 99$). Uraditi ispis svih svih brojeva u rasponu vrijednosti od 10 do n , čiji je zbir cifara manji od njihovog proizvoda. Primjer jednog takvog broja je 23, jer je $2+3 < 2 * 3$

Zadatak 1. X18

Napisati program koji omogućuje korisniku unos prirodnog broja n (gdje je $n \geq 10$). Uraditi ispis svih svih brojeva djeljivih sa 3, čija su zadnje dvije cifre 33, u rasponu vrijednosti od 10 do n .

Zadatak 1.X19

Napisati program koji omogućuje korisniku unos dana, mjeseca i godine. Potrebno je ispisati da li je uneseni datum validan ili ne. Voditi računa o tome da korisnik može unijeti prijestupnu godinu. Zabranjuje se unos negativnih te drugih nevalidnih vrijednosti tipa *dan* > 31 ili *mjesec* > 12.

Metoda za utvrđivanje da li je godina prijestupna je sljedeća:

```
int godina = 2012;
```

```
bool prijestupna = ((godina % 4 == 0) && (godina % 100 != 0)) || (godina % 400 == 0);
```

Zadatak 1.X20

Napisati program koji omogućuje korisniku unos dana, mjeseca i godine za tri različita datuma (ukupno 9 različitih unosa vrijednosti). Program treba da ispiše dan, mjesec i godinu najstarijeg datuma od unesena tri.