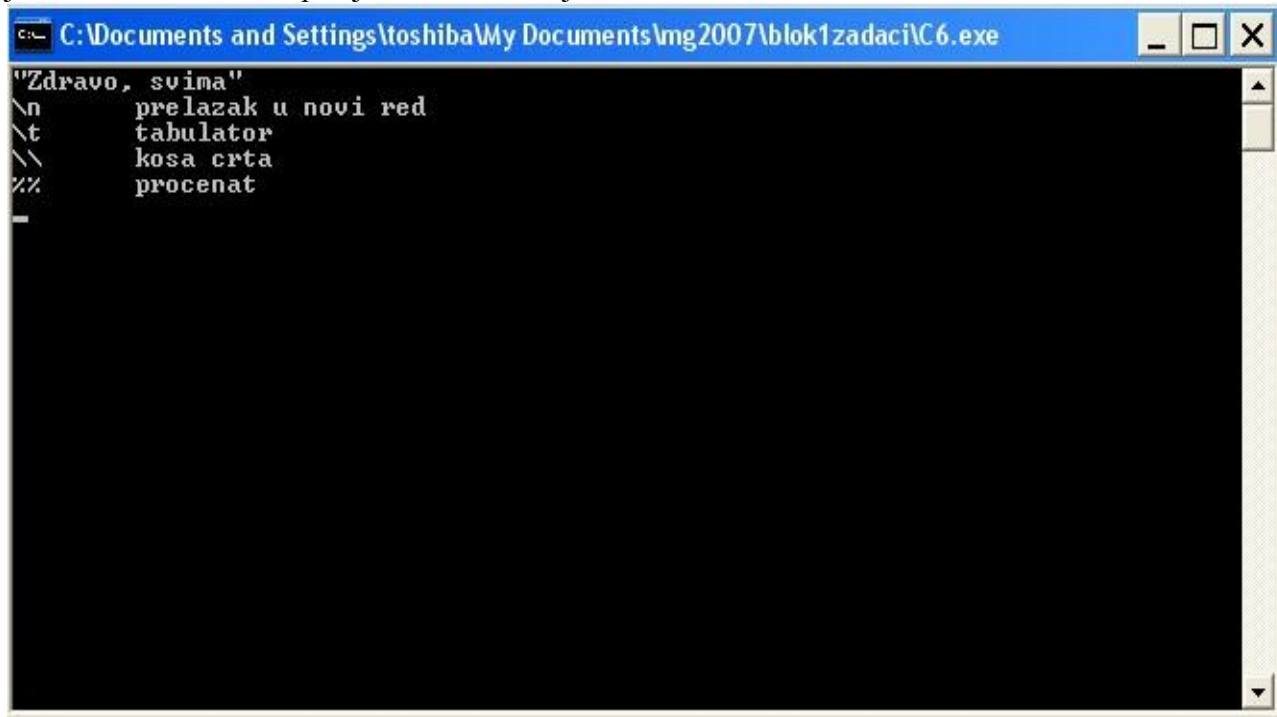


I BLOK - ZADACI

NCP=Napisati C program

1. NCP koji na standardni izlaz ispisuje sledeći sadržaj



```
C:\Documents and Settings\toshiba\My Documents\img2007\blok1zadaci\C6.exe
"Zdravo, svima"
\n      prelazak u novi red
\t      tabulator
\\      kosa crta
%%      procenat
```

2. Sledeći C program za određivanje rešenja linearne jednačine $Ax+B=0$, za A različito od nule, sadrži više grešaka. Ispraviti sve greške

Napomena: format %f se koristi za ispis realnog broja.

```
main
int _x, a, b
{
    scanf('%d, %d', a, b);
    _x=-b/a;
    printf('%f', _x);
}
```

3. NCP koji učitava 4 cela broja i ispisuje na standardni izlaz zbir prva dva, razliku prvog i trećeg, proizvod sva četiri broja, ostatak pri deljenju zbira prva dva broja razlikom prvog i trećeg broja.

ZADACI ZA OCENJIVANJE

4. (12 poena) NCP koji sa standardnog ulaza učitava prirodan broj i ispisuje cifru desetica tog broja.

ULAZ	IZLAZ
------	-------

158	5
12	1
7	0
108	0

```
#include <stdio.h>
main()
{
    int x,x1=0; /*broj, cifra desetica tog broja */
    printf("Unesite broj: ");
    scanf("%d", &x);
    x1 = (x/10)%10;
    printf("Uneti broj= %d\nCifra desetica= %d\n",x, x1);
}
```

5. (12 poena)NCP koji sa standardnog ulaza učitava prirodan broj i ispisuje taj broj bez cifre desetica.

ULAZ	IZLAZ
158	18
12	2
7	7
108	18

```
#include <stdio.h>
main()
{
    int x,x1=0; /*broj, broj bez cifre desetica */
    printf("Unesite broj: ");
    scanf("%d", &x);
    x1 = (x/100)*10 + (x%10);
    printf("Uneti broj= %d\nNovi broj= %d\n",x, x1);
}
```

6. (12 poena)NCP koji permutuje cifru jedinica i desetica učitanom prirodnom broju i ispisuje permutovani broj na standardni izlaz.

ULAZ	IZLAZ
158	185
12	21

7	70
108	180

```
#include <stdio.h>
main()
{
    int x,x1; /*uneseni broj, permutovani broj*/
    printf("Unesi zeljeni broj: ");
    scanf("%d", &x);
    x1 = ((x/100)*10+(x%10))*10 + ((x/10)%10);
    printf("Uneseni broj= %d\nPermutovani broj: %d\n",x,x1);
}
```

7. (10 poena) NCP koji učitava karakter sa tastature i ispisuje na standardni izlaz prethodni i sledeći karakter (gledano u tabeli ASCII kôdova).

ULAZ	IZLAZ
A	prethodni @ sledeci B
b	prethodni a sledeci c
3	prethodni 2 sledeci 4

```
#include <stdio.h>
main()
{
    char karakter;

    printf("Unesi karakter sa tastature\n"); scanf("%c",&karakter);
    printf("Prethodnik= %c\n",karakter-1);
    printf("Sledeci = %c\n",karakter+1);
}
```

8. (12 poena) NCP kojim se određuje koliko vode treba dodati u **X** grama **p1**%-nog rastvora soli da bi se dobio **p2**%-ni rastvor (**p1 > p2**). Celi brojevi X, p1, p2 se unose sa standardnog ulaza.

ULAZ	IZLAZ
X=100 p1=20 p2=5	Treba dodati 300.000000 grama vode
X=5 p1=6 p2=3	Treba dodati 5.000000 grama vode
X=100 p1=5 p2=4	Treba dodati 25.000000 grama vode

Ako treba dodati q grama vode onda važi da: $\frac{p1}{100} * X = \frac{p2}{100} * (X + q)$

```
#include <stdio.h>
main()
{
    int x, p1, p2;
    printf("Unesite X, p1, p2 ");
    scanf("%d%d%d", &x, &p1, &p2);
    printf("Treba dodati %f grama vode\n", (x)*((p1+0.0)/p2-1));
}
```

9. (12 poena) Marko bi sam završio neki posao za **X** sati. Ako bi mu Janko pomogao tako da zajedno rade **Y** sati posao bi se završio za **Z** sati. NCP koji određuje za koje vreme bi Janko sam završio posao. Celobrojne promenljive X, Y, Z se učitavaju sa standardnog ulaza.

ULAZ	IZLAZ
X=5 Y=16 Z=1	Broj sati je 20.000000
X=20 Y=5 Z=15	Broj sati je 20.000000

Marko je sposoban da za jedan sat uradi $\frac{1}{x}$ deo posla.

Janko je sposoban da za jedan sat uradi $\frac{1}{t}$ deo posla.

Za y sati rada, Marko i Janko urade $y * (\frac{1}{x} + \frac{1}{t})$ delova posla. Preostalo je z-y sati koje bi Marko bi odradio za $\frac{(z-y)*1}{x}$

Dakle, važi da $y * (\frac{1}{x} + \frac{1}{t}) + \frac{(z-y)*1}{x} \Rightarrow t = \frac{xy}{(x-z)}$

```
#include <stdio.h>
main()
{
    float x, y, z;
    scanf("%f%f%f", &x, &y, &z);
    printf("Vreme za koje bi Janko uradio posao je %f sati\n", x*y/(x-z));
}
```

10.(10 poena) Trougao se formira od prirodnih brojeva 1, 2, 3, ... na sledeći način:

1
2 3 4

5 6 7 8 9

.....

NCP kojim se određuje suma i -tog reda, gde se prirodan broj i učitava sa standardnog ulaza.

Za svako $i \geq 1$, može se uočiti da red počinje brojem $j=(i-1)^2+1$, a završava se brojem $n=i^2$.

Dakle treba sabrati brojeve od j do n , gde važi da $\sum_j^n k = \sum_1^n k - \sum_1^{j-1} k$

Znajući da $\sum_j^n k = \frac{n*(n+1)}{2}$ važi da $\sum_j^n k = \frac{n*(n+1)}{2} - \frac{(j-1)*(j)}{2}$

Kako je $j=(i-1)^2+1$, $n=i^2$, onda $\frac{n*(n+1)}{2} - \frac{(j-1)*(j)}{2} = \frac{i^2*(i^2+1)}{2} - \frac{((i-1)^2)*((i-1)^2+1)}{2} = 2*i^3 - 3*i^2 + 3*i - 1$

```
#include <stdio.h>
main()
{
    int i;
    scanf("%d", &i);
    printf("Suma reda %d je %d\n", i, 2*i*i*i-3*i*i+3*i-1);
}
```

Ocene: 5(70-80), 4(55-69), 3(35-55), 2(20-34).