

NCP koji će sa standardnog ulaza (do markera kraja) prebrojati i ispisati na standardni izlaz ukupan broj karaktera, broj prelazaka u novi red i broj reči.

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

```
main()
```

```
{
```

```
    int znak;
```

```
    long linije=0, br_znak=0;
```

```
    while ((znak=getchar()) != EOF)
```

```
        { br_znak++;
```

```
          if (znak=='\n') linije ++;
```

```
        }
```

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

```
}
```

Za domaći prepraviti program tako da broji reči na ulazu.

NCP koji će u tekstu sa standardnog ulaza prebrojati koliko puta se pojavila svaka cifra dekadnog brojnog sistema, koliko puta su se pojavile beline (blanko, horizontalni tab, prelaz u novi red) i koliko puta su se pojavili ostali znaci.

```
#include <stdio.h>
main()
{
    int znak, br, White, Ostali, cifra[10];
    White=Ostali=0;
    for(br=0; br<10; ++br) cifra[br]=0;
    while( (znak=getchar())!=EOF )
        if( znak >='0' && znak<='9' ) ++cifra[znak-'0'];
        else if ( znak==' ' || znak=='\n' || znak=='\t' ) ++White;
        else ++Ostali;
    for( br=0; br<10; ++br )
        printf( "Cifra %d se pojavljuje %d puta\n", br,cifra[br] );
    printf("\nBelina ima: %d Ostalih: %d\n", White, Ostali);
}
```

NCP koji unosi broj sa standardnog ulaza, formira broj sa ciframa u obrnutom poretku i ispisuje ga na standardni izlaz.

```
#include <stdio.h>
main() {
    int n,t=0;
    printf("Unesite broj\n");
    scanf("%d",&n);
    do
    {
        t=t*10+n%10;
        n/=10;
    }while(n);
    printf("Novi broj je %d\n", t);
    return 0;
}
```

Primer ilustruje inicijalizaciju nizova.

```
#include <stdio.h>
main() {
/* Niz inicijalizujemo tako sto mu navodimo vrednosti u viticasnim
zagradama.
    Dimenzija niza se odredjuje na osnovu broja inicijalizatora */
    int a[] = {1, 2, 3, 4, 5, 6};
/* Isto vazi i za niske karaktera */
    char s[] = {'a', 'b', 'c'};
/* Ekvivalentno prethodnom bi bilo char s[] = {97, 98, 99}; */
/* Broj elemenata niza */
    int a_br_elem = sizeof(a)/sizeof(int);
    int s_br_elem = sizeof(s)/sizeof(char);
/* Ispisujemo nizove */
    int i;
    for (i = 0; i < a_br_elem; i++) printf("a[%d]=%d\n",i, a[i]);
    for (i = 0; i < s_br_elem; i++) printf("s[%d]=%c\n",i, s[i]);
}
```

[NCP koji ispisuje tablicu množenja do 10.]

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

```
int main() {
```

```
    int i,j;
```

```
    for(i=1; i<=10; i++) {
```

```
        for(j=1; j<=10; j++) printf("%4d", i*j);
```

```
        printf("\n");
```

```
    }
```

```
}
```

NCP koji ispisuje prvih n prostih brojeva

```
#include<stdio.h>
int prost(int);
main() {
    int n, i, br;
    printf("Unesite koliko prostih brojeva zelite da dobijete: \n");
    scanf("%d", &n);
    i = 0; br = 2;
    while(i < n) {
        if (prost(br)) {
            printf("Broj %d je prost.\n", br);
            i++;
        }
        br++;
    }
}
```

NCP koji ispisuje prvih n prostih brojeva

```
int prost(int n) {  
    int prost,i;  
    if (n==1) return 0;  
    prost= (n%2!=0) || (n==2);  
    i=3;  
    while ((prost) && (i*i<=n)) {  
        prost=n%i != 0;  
        i=i+2;  
    }  
    return prost;  
}
```