

**NCP koji proverava da li se na k-tom bitu broja koji se ucitava sa standardnog ulaza nalazi 1. Pozicija k se ucitava sa standardnog ulaza.**

```
#include <stdio.h>
main(){
    int n,k;
    printf("Unesite broj i poziciju koju zelite da proverite:\n");
    scanf("%d %d",&n,&k);
    if ((n&(1 << k))!=0) printf("Bit je 1\n");
    else printf("Bit je 0\n");
}
```

**NCP koji postavlja na k-to mesto 1**

```
#include <stdio.h>
void print_bits(unsigned x);
main(){
    int n,k;
    printf("Unesite broj i poziciju:\n");
    scanf("%d %d",&n,&k);
    printf("Binarno uneseni broj je\n");
    print_bits(n);
    printf("Novi broj je %d\n",(n |(1<<k)));
    printf("Binarno, novi broj je\n");
    print_bits((n |(1<<k)));
}
```

## NCP koji postavlja na k-to mesto 1

```
void print_bits(unsigned x){
    unsigned wl = sizeof(unsigned)*8;
    unsigned mask;
    for (mask = 1<<wl-1; mask; mask >>= 1)
        putchar(x&mask ? '1' : '0');
    putchar('\n');
}
```

## NCP koji proverava korektnu uparenost zagrada '(' i ')' u tekstu koji dolazi sa standardnog ulaza.

```
#include <stdio.h>
main() {
    int c, br_otv = 0;
    while((c=getchar()) != EOF) {
        if(c=='(') br_otv++;
        if (c== ')') {
            br_otv--;
            if (br_otv<0) {
                printf("Visak zatvorenih zagrada\n"); break; }
        }
    }
    if (br_otv == 0) printf("Zagrade su u redu\n");
    else if (br_otv >0) printf("Visak otvorenih zagrada\n");
}
```

NCP koji prihvata sa standardnog ulaza pozitivan ceo broj  $n$  ( $n \leq 50$ ), a zatim prihvata po jedan element  $n$ -dimenzionalnog niza celih brojeva. Formirati drugi niz koji sadrži samo nenegativne elemente unetog niza, a potom članove drugog niza ispisati na standardni izlaz.

```
#include <stdio.h>
main() {
int n,indeks,a[50],j,rezultat[50];
do {
printf("Unesite broj elemenata niza\n");
scanf("%d", &n);
} while (n < 1 || n > 50);
for( indeks=0; indeks<n; indeks++) scanf("%d", &a[indeks]);
for( indeks=0, j=0; indeks<n; indeks++) {
if(a[indeks]<0) continue;
rezultat[j++]=a[indeks];
}
printf("\nNovi niz je: ");
for( indeks=0; indeks<j; indeks++) printf("%d\t", rezultat[indeks]);
printf("\n");
}
```

NCP koji unosi realan broj sa standardnog ulaza i ispisuje na standardni izlaz: taj broj, najmanji ceo broj ne manji ( $\geq$ ) od tog broja, najveći ceo broj ne veći ( $\leq$ ) od tog broja, kvadratni koren tog broja (ako broj je nenegativan), kub tog broja, kosinus tog broja.

```
#include <stdio.h>
#include <math.h>
main()
{ double x;
printf("Unesite broj: ");
scanf( "%lf", &x);
printf("\n\nUneli ste: %lf", x);
printf("\nCeil Vaseg broja: %lf", ceil(x));
printf("\nFloor Vaseg broja %lf", floor(x));
if( x >= 0 ) printf("\nKvadratni koren: %lf", sqrt(x) );
else printf("\nNegativni broj" );
printf("\nKub Vaseg broja %lf", pow(x,3));
printf("\nCosinus: %lf\n", cos(x));
}
```

**gcc -o zad.c rac -lm**

## Enumeracija i enumerisane konstante

```
enum boolean {NO, YES};
enum meseci {JAN = 1, FEB, MAR, APR, MAJ, JUN, JUL,
AVG, SEP, OKT, NOV, DEC}
enum boje {CRVENA, ZELENA=5, PLAVA, LJUBICASTA=10,
ZUTA, CRNA}
```

```
koriscenje: int x=0; boje b;
x=CRVENA+3; /*x ce biti jednako tri*/
b=ZELENA;
x=b+CRNA; /* 5 + 12=17*/
b=0; /*Greska, ovako ne moze!!!*/
```

NCP koji tekst sa standardnog ulaza prepisuje na standardni izlaz pretvarajući početna slova rečenice u velika. Pretpostaviti da se u tekstu znaci .?! pojavljuju samo kao znaci završetka rečenica. Dakle, u tekstu nema rednih brojeva koji bi se završavali tačkom. Tekst se završava markerom kraja datoteke.

```
#include <stdio.h>
#include <ctype.h>
main() {
enum {F,T};
int znak,prvi=T;
while ( (znak=getchar()) != EOF) {
if (isupper(znak)) {
if (!prvi) znak=tolower(znak);
else prvi=F; }
else if (islower(znak)) {
if(prvi) {
znak=toupper(znak);
prvi=F; }
}
else
if (znak == '.' || znak == '!' || znak == '?') prvi=T;
putchar(znak);
}
```