

# Zadaci za vježbu za 2.MI

# Kad bi pogađali...

- petlje
- znakovni nizovi, izmjena sadržaja znakovnog niza, '\0'
- 2d polja (matrice)
- operatori posmaka, gets, algoritam za računanje prostog broja

# Kako rješavati zadatak

- Pažljivo pročitati zadatak
- Pažljivo pogledati primjera ako postoje – ponekad je zadatak dodatno definiran primjerom
- Rubni uvjeti (prazan zn.niz, zn.niz duljine 1 , matrica 1x1 i sl.)
- Probati na papiru
- Struktura programa

# Primjer

Napisati program koji će s tipkovnice učitavati ocjene.

Učitavanja prekinuti ukoliko unesena ocjena nije u intervalu [1,5].

Ispisati koliko je bilo uneseno jedinica, dvojki, trojki, četvorki i petica.

**Npr.**

**Unesite ocjenu:1**

**Unesite ocjenu:2**

**Unesite ocjenu:3**

**Unesite ocjenu:4**

**Unesite ocjenu:5**

**Unesite ocjenu:1**

**Unesite ocjenu:2**

**Unesite ocjenu:7**

**1=2**

**2=2**

**3=1**

**4=1**

**5=1**

# Analiza

Možemo prijaviti variijable:

```
int br1, br2, br3, br4, br5;
```

Bolje:

```
int br[5];
```

# Rješenje

```
#include <stdio.h>

int main(){
    int ocjena, i;
    int brojac[5] = {0};

    do {
        printf("\nUnesite ocjenu:");
        scanf("%d", &ocjena);
        if (ocjena >= 1 && ocjena <= 5){
            brojac[ocjena-1]++;
        }
    } while (ocjena >= 1 && ocjena <= 5);
    for (i=0; i<5; ++i){
        printf("\n%d=%d", i + 1, brojac[i]);
    }
    return 0;
}
```

# Primjer

Napisati program koji učitava cijele brojeve. Učitavanje treba prekinuti kada se učitava broj 0 (taj se broj ne uzima u obzir). Potrebno je ispisati broj učitanih peteroznamenkastih, četveroznamenkastih, troznamenkastih, dvoznamenkastih i jednoznamenkastih brojeva.

# Analiza

Da li možemo koristiti polje brojača?

Možemo li smisliti izraz

```
br [ funkcija(ucitaniBroj) ] ++ ;
```



# Rješenje

```
#include <stdio.h>
#include <math.h>
int main(){
    int br[5] = {0};
    int broj, i;
    int brZnam;
    while (1) {
        scanf ("%d", &broj);
        if (broj == 0)
            break;
        if (broj < 0)
            broj = -broj;
        if (broj < 100000) {
            brZnam=0;
            while(broj){
                broj/=10;
                ++brZnam;
            }
            br[ brZnam - 1 ]++;
            /* ili: br[ (int) log10(broj) ]++; */
        }
    }
    for (i=4; i>=0; --i){
        printf ("Broj %d-znamenkastih je: %d\n", i+1, br[i]);
    }
    return 0;
}
```

# Primjer

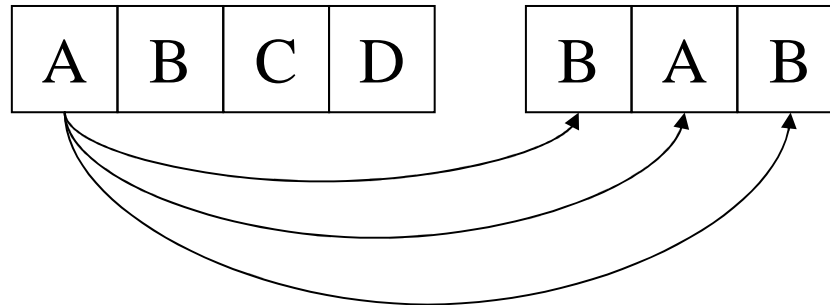
Napisati program koja učitava dva znakovna niza i ispisuje slova koja se nalaze i u jednom i u drugom nizu. Pretpostaviti da nizovi sadrže samo velika slova engleske abecede.

Svako zajedničko slovo treba ispisati samo jedanput.

Primjer: za nizove AABCDA i DAAD treba ispisati AD

# Analiza

- Prijavimo polje brojača – za svako slovo engl.abecede



A	1
B	2
C	0
...	...

# Rješenje

```
#include <stdio.h>
#define MAX 100
int main(){
    char niz1[MAX], niz2[MAX];
    int i, j;
    int brojac['Z'-'A'+1] = {0};

    /* Buduci da u zadatku pise da se unose samo velika slova (a
    razmak NIJE veliko slovo) mozemo koristiti %s.*/

    printf("\nUnesite prvi niz:");
    scanf("%s", niz1);
    printf("\nUnesite drugi niz:");
    scanf("%s", niz2);
```

# rješenje (nastavak)

```
/* Moramo usporedjivati "svaki sa svakim". */
for (i=0; niz1[i] != 0; ++i){
    for (j=0; niz2[j] != 0; ++j){
        if (niz1[i] == niz2[j]){
/* Povecavamo brojac koji odgovara zajednickom slovu */
            brojac[niz1[i]-'A']++;
        }
    }
}

printf("\nZajednicki znakovi su:");
for (i=0; i < 'Z'-'A'+1; ++i){
    if (brojac[i] > 0){
        printf("%c", 'A' + i);
    }
}
return 0;
}
```

## Primjer – učitavanje sve dok...

Napisati program koji će učitati cijeli broj  $n$  ( **$1 \leq n \leq 10$** , *ukoliko unesena vrijednost nije unutar intervala, učitavanje treba ponavljati sve dok se ne unese ispravna vrijednost*) i elemente matrice dimenzija  $n \times n$  te ...

# Analiza

- Matricu definirati na maksimalnu vrijednost koju može poprimiti zadana varijabla:

	Ili još bolje:	<b>Ali NIKAKO:</b>
<code>int m[10][10];</code>	<code>#define MAX 10 int m[MAX][MAX];</code>	<code>int n; int m[n][n];</code>

- Riješiti pomoću do-while petlje:  
do {  
    */\* učitati tražene vrijednosti \*/*  
} while ( !(zadani uvjet) );

# Rješenje:

```
#include <stdio.h>
#define MAX 10
#define MIN 1
int main(){
    int m[MAX][MAX];
    int n;
    do {
        printf("Unesite n (1<=n<=10):");
        scanf("%d", &n);
    } while (!(n >= MIN && n <= MAX));
    printf ("\nOK.");
}
```



# Primjer

Napisati program koji će učitati cijeli broj  $n$  ( $1 \leq n \leq 10$ , *ukoliko unesena vrijednost nije unutar intervala učitavanje treba ponavljati sve dok se ne unese ispravna vrijednost*) i zatim dvodimenzionalno polje znakova dimenzija  $N \times N$  popuniti naizmjenice s '#' i '\_' kako bi se dobila "šahovnica" (u gornjem lijevom kutu je uvijek '#').

Npr.

#	_	#	_
_	#	_	#
#	_	#	_
_	#	_	#

# Analiza

- gornji lijevi kut uvijek započinjemo s '#'
- da li se može izračunati kakva formula s obzirom na indekse matrice?

	0	1
0	#	—
1	—	#

	0	1	2
0	#	—	#
1	—	#	—
2	#	—	#

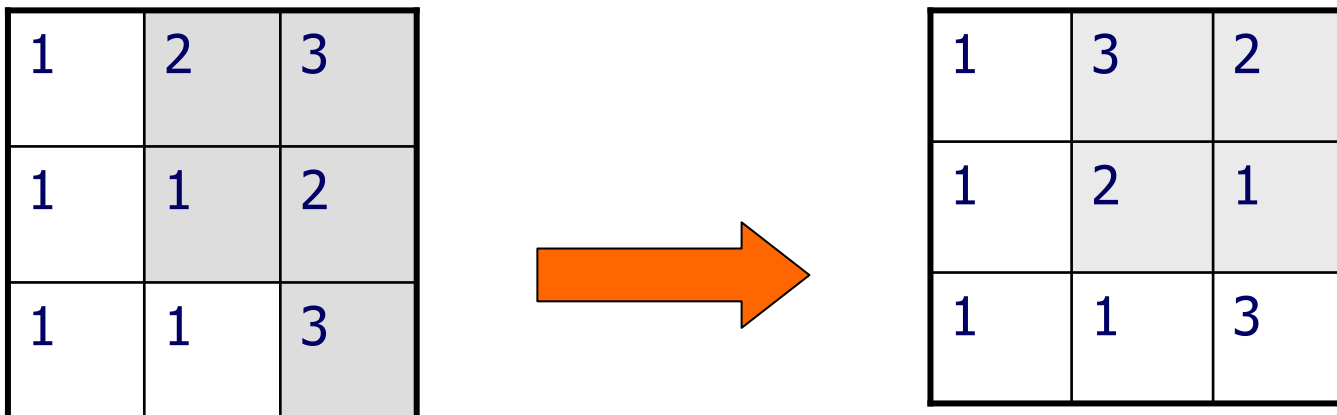
# Rješenje

```
#include <stdio.h>
#define MAX 10
#define MIN 1
int main(){
    char m[MAX][MAX];
    int n, i, j, zn;
    do {
        printf("Unesite n (1<=n<=10):");
        scanf("%d", &n);
    } while (!(n>=1 && n<=10));
    for (i=0; i<n; ++i){
        for (j=0; j<n; ++j){
            m[i][j] = ( (i+j) % 2 == 0) ? '#' : '_';
            printf("%c ", m[i][j]);
        }
        printf("\n");
    }
}
```

# Primjer

Napisati program koji će učitati cijeli broj  $n$  ( $1 \leq n \leq 10$ ). Ukoliko unesena vrijednost nije unutar intervala, prekinuti izvođenje programa. U suprotnom učitati vrijednost cjelobrojne matrice dimenzija  $n \times n$  i zatim u svakom retku zamijeniti vrijednosti zadnjeg parnog i neparnog člana matrice.

Npr.



# Analiza

- u svakom retku nađemo indeks zadnjeg parnog i zadnjeg neparnog
- ako su oba indeksa valjana – zamijenimo vrijednosti
- kako ćemo znati da li su valjani?  
-> inicijalno postavimo indekse na -1

## Rješenje:

```
#include <stdio.h>
#define MAX 10
#define MIN 1
int main(){
    int m[MAX][MAX];
    int n, i, j, iZp, iZn, pom;

    printf("Unesite n (1<=n<=10):");
    scanf("%d", &n);

    if (n>=MIN && n<=MAX) {
        printf ("Unesite matricu: \n");
        for (i=0; i<n; ++i){
            for (j=0; j<n; ++j){
                scanf("%d", &m[i][j]);
            }
            printf("\n");
        }
        ...
    }
```

```

...
    for (i=0; i<n; ++i){
        iZp = iZn = -1;
        for (j=0; j<n; ++j){
            if (m[i][j] % 2 == 0) {
                iZp = j;
            } else {
                iZn = j;
            }
        }
        if (iZp>=0 && iZn >=0) {
            pom = m[i][iZp];
            m[i][iZp] = m[i][iZn];
            m[i][iZn] = pom;
        }
    }
    for (i=0; i<n; ++i){
        for (j=0; j<n; ++j){
            printf("%d ", m[i][j]);
        }
        printf("\n");
    }
}
}

```

Ili bolje(trajimo zdesna na lijevo i prekidamo  
čim pronađemo parni i neparni):

...

```
    for (i=0; i<n; ++i){  
        iZp = iZn = -1;  
        for (j=n; j>=0; --j){  
            if (m[i][j] % 2 == 0 && iZp==  
1) {  
                iZp = j;  
            } else if (m[i][j] % 2 == 1 &&  
iZn==-1) {  
                iZn = j;
```



# Primjer

- Napisati program koji će učitati znakovni niz i ispisati da li je učitani znakovni niz palindrom.

- Npr.

pero

pero nije palindrom.

ana

ana je palindrom.

# Analiza

0	1	2	3
a	b	b	a

$$d = 4, d/2 = 2$$

0	1	2
a	n	a

$$d = 3, d/2 = 1$$

# Rješenje

```
#include <stdio.h>
int main(){
    char niz[512+1];
    int i=0;
    int d, p=1;
    gets(niz);
    d=0;
    while (niz[d]){
        ++d;
    }
    for (i=0; i<d/2; ++i){
        if (niz[i] != niz[d-i-1]){
            p=0;
            break;
        }
    }
    printf ("Niz %s %s palindrom", niz, (p)? "JE" : "NIJE" );
    return 1;
}
```

## Primjer – stvaranje novog niza

- Napisati program koji će učitati s tipkovnice jednu rečenicu (možete pretpostaviti da korisnik neće unijeti više od 512 znakova). Ispisati rečenicu i "ispravljenu rečenicu" koja nastaje tako da se iza svakog zareza umetne razmak ukoliko nije postojao.

Npr.

Ako je uneseno: tko zna,zna.

Treba ispisati:

Originalna recenica: tko zna,zna.

Ispravljena recenica: tko zna, zna.

# Analiza

- Nije rečeno da se mora **promijeniti** zadani niz, štoviše traži se da se i on ispiše što znači da je sačuvan. Dakle, možemo koristiti drugi niz u koji ćemo upisati "ispravljenju" rečenicu.
- Što da je trebalo promijeniti zadani niz?

# Rješenje

```
#include <stdio.h>
int main(){
    char r[512+1];
    char ir[2*512+1];
    int i, j;
    printf("Unesite recenicu:\n");
    gets(r);
    for (i=0, j=0; r[i] != 0; ++i){
        if (r[i] == ',' && r[i+1] != ' '){
            ir[j++] = ',';
            ir[j++] = ' ';
        } else {
            ir[j++] = r[i];
        }
    }
    ir[j] = 0;  // !!
    printf("\nOriginalna recenica:%s", r);
    printf("\nIspravljena recenica:%s", ir);
    return 0;
}
```

## Primjer – skraćivanje niza

- Napisati program koji će učitati s tipkovnice jednu rečenicu (možete pretpostaviti da korisnik neće unijeti više od 512 znakova). Iz rečenice izbacite sve samoglasnike (bez obzira na velika i mala slova).

Npr.

Ako je uneseno: TKO zna,zna.

Treba ispisati: TK zn,zn.

# Analiza

- Ovdje je potrebno izmijeniti zadani niz
- Prepisivati ćemo niz samu sebe, preskačući samoglasnike
- Dvije varijable: i – trenutni indeks starog niza  
j – trenutni indeks novog niza
- Postaviti oznaku kraja niza!!!



# Rješenje

```
#include <stdio.h>
int main(){
    char r[512+1];
    char zn;
    int i, j;
    printf("Unesite recenicu:\n");
    gets(r);
    for (i=0, j=0; r[i] != 0; ++i){
        if (r[i] >= 'a' && r[i]<='z'){
            zn = r[i] - ('a'-'A');
        } else {
            zn = r[i];
        }
        if (!(zn=='A' || zn=='E' || zn=='I' || zn=='O' || zn== 'U')){
            r[j] = r[i];
            ++j;
        }
    }
    r[j] = 0;  // !!
    printf("\nRecenica bez samoglasnika:%s", r);
    return 0;
}
```

## Primjer – proširivanje niza

- Napisati program koji će učitati s tipkovnice jednu rečenicu (možete pretpostaviti da korisnik neće unijeti više od 512 znakova). U sredinu rečenice stavite znak 'X'.

Npr.

Ako je uneseno: AB

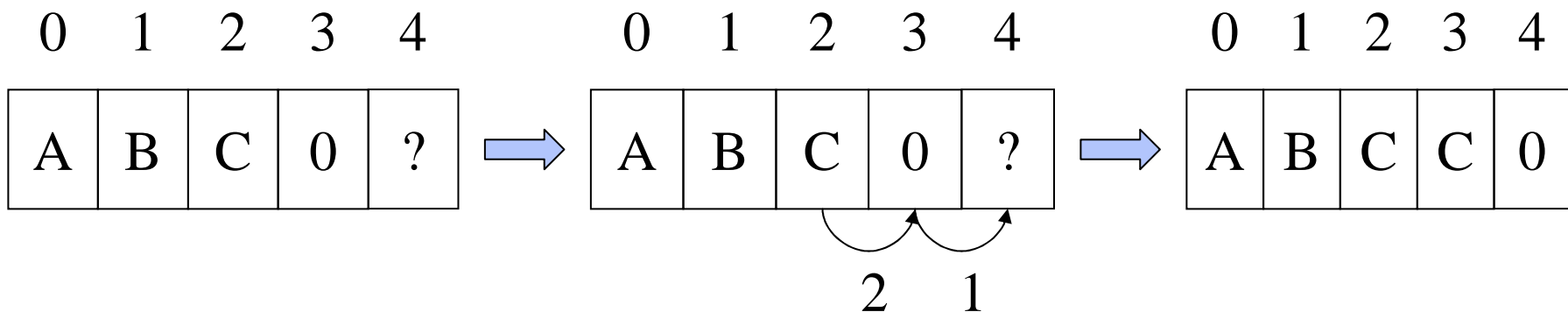
Treba ispisati: AXB

Ako je uneseno: ABC

Treba ispisati: ABXC

# Analiza

- Ovdje je potrebno izmijeniti zadani niz
- Iz primjera vidimo što se smatra "sredinom":
  - d – duljina niza
  - ako je d paran dodati na indeksu  $d/2$
  - ako je d neparan dodati na indeksu  $d/2+1$*(paziti na rubne uvjete, što ako je  $d=0$ )*
- PRIJE DODAVANJA ZNAKA POTREBNO JE NAPRAVITI MJESTO:  
pomaknuti sve znakove(iza mjesta gdje mislimo dodati) za jedno mjesto.



što ako bi kopirali drugim redoslijedom?

# Rješenje

```
#include <stdio.h>

int main(){
    char r[512+1+1];
    int i, idxNovi, d;

    printf("Unesite recenicu:\n");
    gets(r);

    d=0;
    while (r[d]){
        ++d;
    }

    idxNovi = ( d%2 == 0) ? d/2 : d/2+1; /* ili idxNovi = d/2 + d%2;*/
    for (i=d+1; i>idxNovi; --i){
        r[i] = r[i-1];
    }
    r[idxNovi] = 'X';

    printf("\n%s", r);
    return 0;
}
```

# Primjer

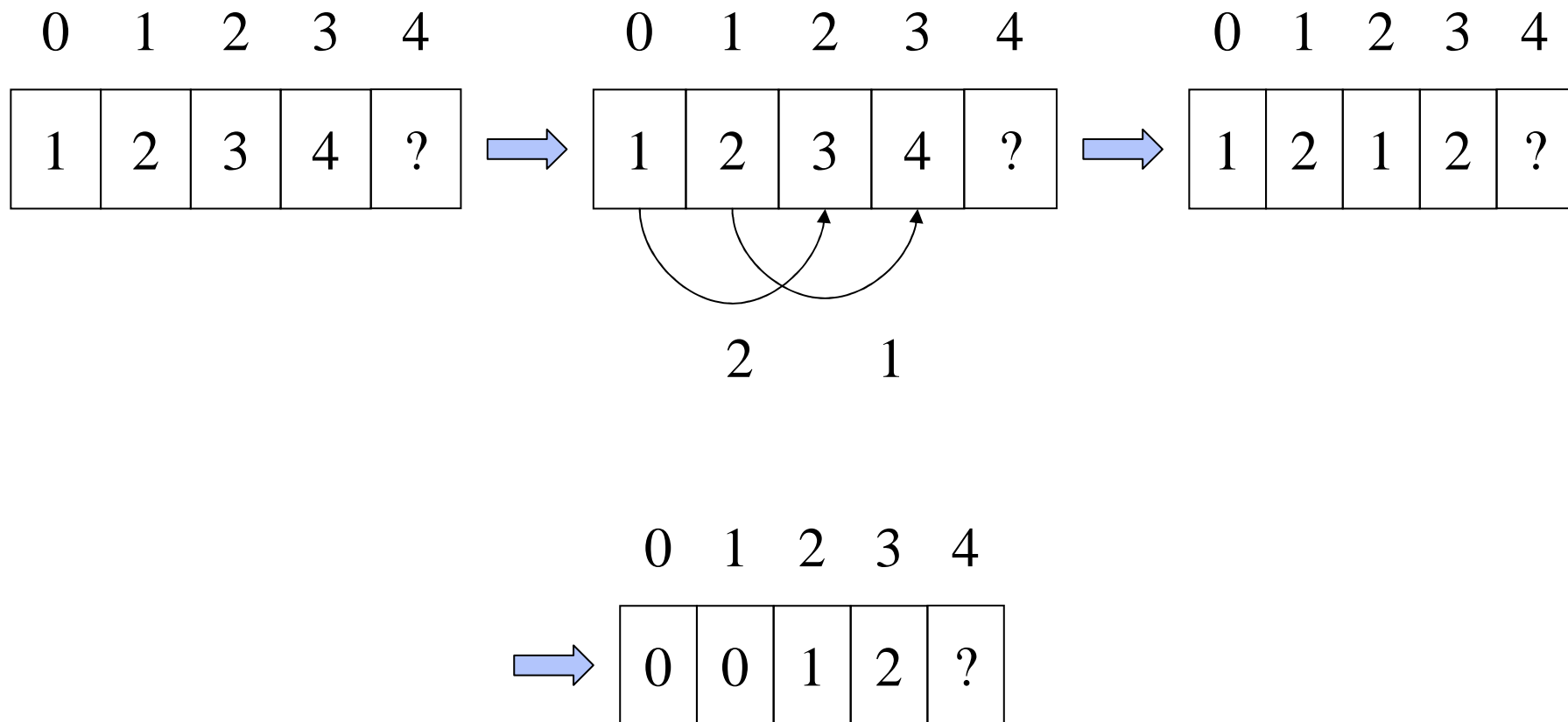
Napisati program koji će sve elemente zadanog cjelobrojnog polja p od N elemenata posmaknuti udesno za određeni broj mjesta posmak ( $0 \leq \text{posmak} < N$ ). Ispražnjeni elementi pune se nulama.

Primjer posmaka za posmak = 2 i polje [10 40 50 60 12]

Polje prije posmaka: [10 40 50 60 12]

Polje nakon posmaka: [0 0 10 40 50]

# Analiza



# Rješenje

```
#include <stdio.h>
#define N 100
int main(){
    int p[N], posmak, brojelem, i;
    do{
        printf("Unesite broj elemenata polja: ");
        scanf("%d", &brojelem);
    } while (!(brojelem>0 && brojelem<N));
    printf("Unesite clanove polja: ");
    for (i=0; i<brojelem; i++ ){
        scanf("%d",&p[i]);
    }
    do {
        printf("Unesite posmak: ");
        scanf("%d", &posmak);
    } while (!(posmak>0 && posmak<N ));

    for (i = N - 1 - posmak; i >= 0; i--){
        p[i + posmak] = p[i];
    }
    for (i = posmak - 1; i >= 0; i --){
        p[i] = 0;
    }
    printf("\n");
    for (i=0; i<brojelem; i++ ){
        printf("%d ", p[i]);
    }
    return 0;
}
```

## 5. primjer

Napisati program koji učitava dimenzije cjelobrojne matrice (broj redaka  $\leq 10$ ; broj stupaca  $\leq 10$ ), te članove matrice, a zatim ispisuje matricu, te broj redaka matrice koji imaju sljedeće svojstvo: svaki element retka (osim prvoga) je za jedan veći od prethodnog elementa u retku.

```
#include <stdio.h>
#define MAXRED 10
#define MAXSTUP 10

int main () {
    int mat[MAXRED][MAXSTUP];
    int brRed, brStup;

    int brojDobrihRedaka = 0;
    int dobarRedak;
    int i, j;
    do {
        /* učitati dimenzije matrice */
        scanf("%d %d", &brRed, &brStup);
        while (brRed<1 || brRed>10 || brStup<1 || brStup>10)
            /* while (!(brRed>=1 && brRed<=10 && brStup>=1 && brStup<=10)) */
    }
```



## 5. primjer (nastavak)

```
/* učitati elemente matrice */
for (i = 0; i < brRed; i++) {
    for (j = 0; j < brStup; j++) {
        scanf("%d", &mat[i][j]);
    }
}

/* provjeriti za svaki redak ispunjava li zadani uvjet */
for (i = 0; i < brRed; i++) {
    dobarRedak = 1;
    /* je li j. clan za 1 veci od (j-1). clana */
    for (j = 1; j < brStup && dobarRedak == 1; j++) {
        if (mat[i][j] != mat[i][j - 1] + 1) {
            dobarRedak = 0;
        }
    }
    /* ako su svi clanovi retka ispunili uvjet, uvecaj brojac za 1 */
    if (dobarRedak == 1) {
        ++ brojDobrihRedaka;
    }
    dobarRedak = 1;
}
```

## 5. primjer (nastavak)

```
/*ispis matrice */
for (i = 0; i < brRed; i++) {
    for (j = 0; j < brStup; j++) {
        printf("%d ", mat[i][j]);
    }
    printf("\n");
}

printf("Broj redaka: %d", brojDobrihRedaka);
return 0;
}
```

## 6. primjer

Napisati program koji učitava cijeli broj koji nema više od 9 znamenki. Ako broj ima 5 znamenki, treba ispisati umnožak znamenki. Ako broj ima 4 znamenke, treba ih redom oduzeti (prva znamenka-druga znamenka- treća znamenka-četvrta znamenka). Ako broj ima 3 znamenke, treba ispisati broj bez središnje znamenke. Ako broj ima 2 znamenke, treba ispisati broj koji čine obrnute znamenke učitano broj. Ako broj ima 1 znamenku, treba ispisati kvadrat te znamenke. Ako broj ima više od šest znamenki, potrebno je ispisati prve dvije znamenke učitano broj.

Napomena: zadatak riješiti bez korištenja if naredbe.

```
#include <stdio.h>

int main()
{
    int broj, znamenke[9]={0}, i=0, brZnam=0, produkt=1;
    scanf("%d", &broj);
    /*na mjestu i u polju znamenke je znamenka na mjestu i u učitano broj*/
    do{
        znamenke[i]=broj%10;
        broj=broj/10;
        i++;
        brZnam++;
    } while (broj!=0);
```

## 6. primjer (nastavak)

```
switch (brZnam)
{
    case 5:
        for (i=0;i<5; i++) produkt*=znamenke[i];
        printf("%d", produkt);
        break;

    case 4:
        broj = znamenke[3]-znamenke[2]-znamenke[1]-znamenke[0];
        printf ("%d", broj);
        break;

    case 3:
        broj = znamenke[2]*10 + znamenke[0];
        printf ("%d", broj);
        break;
```

## 6. primjer (nastavak)

```
case 2:
    broj = znamenke[0]*10 + znamenke[1];
    printf ("%d", broj);
    break;

case 1:
    printf ("%d",znamenke[0]*znamenke[0]);
    break;

default:
    printf ("%d %d", znamenke[brZnam-1], znamenke[brZnam -2]);
}

return 0;

}
```

## 7. primjer

Napisati program koji će sve elemente zadanog cjelobrojnog polja p od N elemenata posmaknuti za određeni broj mjesta posmak ( $\text{posmak} < N$ ). Ukoliko je posmak pozitivan, elementi polja posmiču se udesno, a ukoliko je posmak negativan, elementi polja posmiču se ulijevo. Ispraznjeni elementi pune se nulama.

Primjer posmaka za  $\text{posmak} = 2$  i polje [10 40 50 60 12]

Polje prije posmaka: [10 40 50 60 12]

Polje nakon posmaka: [0 0 10 40 50]

```
#include <stdio.h>
#define N 100

int main()
{
    int p[N], posmak, brojelem, i;

    do{
        printf("Unesite broj elemenata polja: ");
        scanf("%d", &brojelem);
    } while (brojelem<1 || brojelem>N);
```

## 7. primjer (nastavak)

```
printf("Unesite clanove polja: ");
for (i=0; i<brojelem; i++ ){
    scanf("%d",&p[i]);
}

do{
    printf("Unesite posmak: ");
    scanf("%d", &posmak);
}while (posmak>N || posmak<-N );

/* ako je posmak udesno */
if (posmak > 0) {
    for (i = N - 1 - posmak; i >= 0; i --)
        p[i + posmak] = p[i];
    for (i = posmak - 1; i >= 0; i --)
        p[i] = 0;
}
```

## 7. primjer (nastavak)

```
/* ako je posmak ulijevo */
else if (posmak < 0) {
    posmak = -posmak;
    for (i = 0; i <= N - 1 - posmak; i++)
        p[i] = p[i + posmak];
    for (i = N - posmak; i < N; i++)
        p[i] = 0;
}

printf("\n");
for (i=0; i<brojelem; i++){
    printf("%d ", p[i]);
}

}
```



## 7. primjer (nastavak)- rješenje s pokazivačima

```
int main()
{
    int p[N]={0}, posmak, brojelem, i;
    do{
        printf("Unesite broj elemenata polja: ");
        scanf("%d", &brojelem);
    } while (brojelem<1 || brojelem>N);

    printf("Unesite clanove polja: ");
    for (i=0; i<brojelem; i++){
        scanf("%d", p+i);
    }

    do{
        printf("Unesite posmak: ");
        scanf("%d", &posmak);
    } while (posmak>=N);
```

## 7. primjer (nastavak)- rješenje s pokazivačima

```
/* ako je posmak udesno */
if (posmak > 0) {
    for (i = N - 1 - posmak; i >= 0; i --)
        *(p+i+posmak) = *(p+i);
    for (i = posmak - 1; i >= 0; i --)
        *(p+i) = 0;
} /* ako je posmak ulijevo */
else if (posmak < 0) {
    posmak = -posmak;
    for (i = 0; i <= N - 1 - posmak; i++)
        *(p+i) = *(p+i+posmak);
    for (i = N - posmak; i < N; i++)
        *(p+i) = 0;
}
printf("\n");
for (i=0; i<brojelem; i++ ){
    printf("%d ", *(p+i));
}
}
```

# Primjer

Napisati program koja učitava dva cijela broja i ispisuje uniju znamenki učitanih brojeva. Pretpostaviti da brojevi nemaju više od 9 znamenki. Znamenke treba ispisati poredane po veličini uzlazno. Svaku znamenku treba ispisati samo jedanput.

Npr. za brojeve 2482 i 335228, treba ispisati 2 3 4 5 8

```
#include <stdio.h>
#define MAX 100
int main(){
    int prviBroj, drugiBroj, i;
    int brojacZnamenki[10] = {0};
    printf("Ucitajte prvi broj: ");
    scanf("%d", &prviBroj);
    printf("Ucitajte drugi broj: ");
    scanf("%d", &drugiBroj);
```

## 2. primjer (nastavak)

```
int i = 0;
/*
Uzima se znamenka s "kraja" broja uz pomoć modulo operacije
te povećava vrijednost elementa polja koji je na poziciji
vrijednosti uzete znamenke
*/
do{
    brojacZnamenki[prviBroj%10]++;
    prviBroj/=10; /* Zadnja se znamenka miče */
} while(prviBroj>0);

do{
    brojacZnamenki[drugiBroj%10]++;
    drugiBroj/=10;
} while(drugiBroj>0);

for(i=0; i<10; i++){
    if(brojacZnamenki[i]) printf("%d ", i);
}
}
```