

Napomene:

- Savjetuje se navedene zadatke riješiti ubrzo nakon predavanja
- Savjetuje se ne gledati rješenja prije nego se pokuša samostalno riješiti zadatke

12. vježbe uz predavanja

1. S tipkovnice učitati 10 cijelih brojeva i zatim ih ispisati u obrnutom poretku u odnosu na poredak kojim su učitani. **Naputak:** učitavati brojeve i pohranjivati ih u polje, zatim ispisati brojeve čitajući polje od 10. člana prema prvom. Kakve se ovdje petlje trebaju koristiti?
2. S tipkovnice učitati 10 cijelih brojeva u polje, poredak članova **obrnuti unutar** polja. Nakon što se poredak članova u polju promijeni, ispisati novi sadržaj polja. Npr. ako su u polje učitani brojevi
1 3 5 7 9 11 13 15 17 19
njihov redoslijed u polju treba promijeniti tako da su članovi polja sljedeći:
19 17 15 13 11 9 7 5 3 1
3. Učitati pozitivni cijeli broj. Uzastopnim dijeljenjem sa 16 odrediti znamenke ekvivalentnog heksadekadskog broja. Ispisati ekvivalentni heksadekadski broj, ali tako da su znamenke heksadekadskog broja ispisane ispravnim redoslijedom. **Naputak:** znamenke dobivene uzastopnim dijeljenjem ubacivati u polje. Ispisati članove polja obrnutim redoslijedom. Koliko članova mora imati polje pri definiciji da bi se mogla obaviti pretvorba bilo kojeg pozitivnog cijelog broja tipa *unsigned int* (pretpostavlja se da *int* koristi 4 okteta).
4. Cjelobrojno polje je potrebno napuniti s prvih 40 članova Fibonaccijevog niza. **Naputak:** za prva dva člana polja ubaciti 1 i 1. Svaki sljedeći član izračunati na temelju prethodna dva. Uočite da se prethodna dva člana uvijek nalaze u polju, pa nije potrebno koristiti varijable *f0* i *f1* kao u primjeru s predavanja. Kakvu petlju ovdje treba koristiti?
5. Cjelobrojno polje je potrebno napuniti s 500 prostih brojeva. Koristiti činjenicu da je neki broj *x* prost ukoliko nije djeljiv s niti jednim prostim brojem koji je manji od *x*. Nakon punjenja polja ispisati sadržaj polja. **Naputak:** u polje kao prvi član ubaciti 2. Za svaki sljedeći neparni broj ("broj kandidat") ispitivati je li djeljiv s bilo kojim prostim brojem koji se do tada već nalazi u polju prostih brojeva (pri tome uvijek preskočiti prvi član, tj. 2). Vanjskom petljom povećava se "broj kandidat", a unutarnjom petljom se testira djeljivost "broja-kandidata" sa svim prethodno upisanim prostim brojevima osim 1. člana. Kakve se petlje ovdje moraju koristiti?

(za provjeru: petstoti prim broj je 3571)

6. Pomoću funkcije `gets` učitajte s tipkovnice niz znakova koji sigurno nije dulji od 30 znakova. Izračunajte i ispišite koliko u učitanoj nizu ima samoglasnika, koliko suglasnika, a koliko ostalih znakova.

Primjer:

ako korisnik upiše sljedeći niz: `printf("%s", "Ana");` <Enter> na zaslon treba ispisati:

Samoglasnika: 3

Suglasnika: 7

Ostalih: 10

7. S tipkovnice, u obliku dekadskih brojeva, učitati najviše 8 ASCII vrijednosti znakova. Učitavanje prekinuti i onda kada korisnik unese ASCII vrijednost koja ne predstavlja veliko slovo abecede. Učitane vrijednosti pohraniti u polje znakova koje se koristi kao niz znakova. Na kraju, ispisati dobiveni niz znakova pomoću funkcije `printf` i formata `%s`.

Primjeri:

ako korisnik upiše sljedeći niz ASCII vrijednosti:

80 69 82 85 64 80 69 <Enter> na zaslon treba ispisati **PERU**

ako korisnik upiše sljedeći niz ASCII vrijednosti:

65 82 71 69 78 84 73 78 65 66 <Enter> na zaslon treba ispisati

ARGENTIN

8. S tipkovnice učitati cijeli broj `n` koji mora biti iz intervala `[3, 10]`. Učitavanje broja `n` ponavljati dok god ne bude ispravno upisan. Nakon toga s tipkovnice učitati `n` cijelih brojeva i pohraniti ih u polje `ulaz`. U polje `parni` prepisati sve parne vrijednosti, a u polje `neparni` prepisati sve neparne vrijednosti iz polja `ulaz`. Na kraju ispisati članove polja `ulaz`, `parni` i `neparni` (svako polje u svom retku, s članovima međusobno odvojenim prazninom).
9. S tipkovnice učitati niz znakova koji sigurno nije dulji od 100 znakova. Učitati cijeli broj, ako broj predstavlja ispravnu poziciju znaka u nizu, iz niza izbaciti znak na zadanoj poziciji te promijenjeni niz ispisati na zaslon. Inače, ispisati poruku "Neispravna pozicija". Smatra se da se pozicije počinju od 1, tj. prvi znak u nizu (s indeksom nula) jest znak na poziciji 1.

Primjeri:

ako korisnik upiše `Ovo je niz` i poziciju 0, ispisati **Neispravna pozicija**.

ako korisnik upiše `Ovo je niz` i poziciju 12, ispisati **Neispravna pozicija**.

ako korisnik upiše `Ovo je niz` i poziciju 3, ispisati **Ov je niz**.

10. S tipkovnice učitati niz znakova koji sigurno nije dulji od 100 znakova. Učitati cijeli broj, ako broj predstavlja ispravnu poziciju znaka u nizu, u niz na zadanu poziciju umetnuti slovo 'X' te promijenjeni niz ispisati na zaslon. Inače, ispisati poruku "Neispravna pozicija". Smatra se da se pozicije počinju od 1, tj. prvi znak u nizu (s indeksom nula) jest znak na poziciji 1.

Primjeri:

ako korisnik upiše `Ovo je niz` i poziciju 0, ispisati `Neispravna pozicija`.

ako korisnik upiše `Ovo je niz` i poziciju 12, ispisati `Neispravna pozicija`.

ako korisnik upiše `Ovo je niz` i poziciju 3, ispisati `OvXo je niz`.

11. S tipkovnice učitati dva niza znakova pri čemu niti jedan niz nije dulji od 20 znakova. Učitati cijeli broj, ako broj predstavlja ispravnu poziciju znaka u prvom nizu, u prvi niz na zadanu poziciju umetnuti drugi niz, te ispisati (sada promijenjeni) prvi niz. Inače, ispisati poruku "Neispravna pozicija". Smatra se da se pozicije počinju od 1, tj. prvi znak u nizu (s indeksom nula) jest znak na poziciji 1.

Primjer:

ako korisnik upiše nizove

`Ovo je niz`

`Novi niz`

i poziciju 0, ispisati `Neispravna pozicija`.

ako korisnik upiše nizove

`Ovo je niz`

`Novi niz`

i poziciju 3, ispisati `OvNovi nizo je niz`.

12. S tipkovnice učitati dva niza znakova pri čemu niti jedan niz nije dulji od 20 znakova. U treći niz znakova upisati rezultat operacije nadovezivanja nizova (konkatenacije nizova) te ispisati treći niz.

Primjer:

ako korisnik upiše nizove

`Ovo je niz`

`Novi niz`

treba formirati i ispisati treći niz čiji je sadržaj `Ovo je nizNovi niz`.

13. Pomoću funkcije gets s tipkovnice učitati niz znakova koji sigurno nije dulji od 80 znakova. Za svako malo slovo engleske abecede (znakovi a-z) ispisati koliko se puta pojavilo u učitanoj nizu.

slovo 'a' pojavilo se xx puta

slovo 'b' pojavilo se xx puta

slovo 'c' pojavilo se xx puta

...

slovo 'z' pojavilo se xx puta

Rješenja

Rješenje 1. zadatka

```
#include <stdio.h>

#define MAXCLAN 10
int main(void) {
    int i;
    int broj[MAXCLAN];

    for (i = 0; i < MAXCLAN; ++i)
        scanf("%d", &broj[i]);

    for (i = MAXCLAN-1; i >= 0; --i)
        printf("%d\n", broj[i]);

    return 0;
}
```

Rješenje 2. zadatka

```
#include <stdio.h>

#define MAXCLAN 10
int main(void) {
    int i;
    int broj[MAXCLAN];
    int pom;

    for (i = 0; i < MAXCLAN; ++i)
        scanf("%d", &broj[i]);

    for (i = 0; i < MAXCLAN/2; ++i) {
        pom = broj[i];
        broj[i] = broj[MAXCLAN-1-i];
        broj[MAXCLAN-1-i] = pom;
    }

    for (i = 0; i < MAXCLAN; ++i)
        printf("%d\n", broj[i]);

    return 0;
}
```

Rješenje 3. zadatka

```
#include <stdio.h>

#define MAXZNAM 8
int main(void) {
    int i;
    unsigned int broj;
    char znamenke[MAXZNAM];
    int brojZnam = 0;
    char znamenka;

    scanf("%u", &broj);

    while (broj > 0) {
        znamenka = broj % 16;
        /* sad pretvoriti broj 0-15 u znak '0'-'9' ili 'A'-'F'
           i ubaciti ga u polje*/
        if (znamenka <= 9)
            znamenke[brojZnam] = '0' + znamenka;
        else
            znamenke[brojZnam] = 'A' + znamenka - 10;

        broj /= 16;
        ++brojZnam;
    }

    for (i = brojZnam-1; i >= 0; --i)
        printf("%c", znamenke[i]);

    printf("\n");
    return 0;
}
```

Pretpostavilo se da će SIGURNO biti učitani cijeli broj veći od nule. Varijabla brojZnam sadrži podatak o tome koliko heksadekaderski broj ima znamenaka. Heksadekaderski broj pohranjen u 4 bajta ima najviše 8 znamenaka.

S obzirom da heksadekaderske znamenke ne mogu biti prikazane znamenkama 0-9, koristi se polje znakova. Postupkom dijeljenja prvo se dobiva znamenka u rasponu 0-15, a zatim se ona transformira u ASCII vrijednost odgovarajućeg znaka ('0' ... '9' i 'A' ... 'F'). Ovakva transformacija ne bi bila nužna kada bi se broj pretvarao u broj s bazom ≤ 10 . Tada bi se znamenke mogle pohraniti kao "obični" cijeli brojevi, npr. short, a kod ispisa svake znamenke bi se koristio format %d. Možete testirati npr. s oktalnim brojevnim sustavom.

Problem se mogao riješiti i tako da se znamenke heksadekaderskog broja u polje ubacuju "odostrag". Tada bi se članovi polja mogli ispisati redom ("odnaprijed"), s time da bi trebalo paziti od kojeg člana polja se počinje s ispisom. Alternativno, svi članovi polja bi se mogli na početku napuniti znakom '0', a onda heksadekaderski broj ispisivati s vodećim nulama. Pokušajte sami.

Rješenje 4. zadatka

```
#include <stdio.h>

#define MAXCLAN 40
int main(void) {
    int i;
    int fbroj[MAXCLAN];
    fbroj[0] = fbroj[1] = 1;

    for (i = 2; i < MAXCLAN; ++i)
        fbroj[i] = fbroj[i-1] + fbroj[i-2];

    for (i = 0; i < MAXCLAN; ++i)
        printf("%d\n", fbroj[i]);
    return 0;
}
```

Rješenje 5. zadatka

```
#include <stdio.h>

#define MAX 500
#define TRUE 1
#define FALSE 0

int main(void) {
    int prosti[MAX];
    int slobodanInd, indProstog, kandidat, jeProst, i;
    prosti[0] = 2;
    slobodanInd = 1;
    kandidat = 3;
    while (slobodanInd < MAX) {
        jeProst = TRUE;
        for (indProstog = 1; indProstog < slobodanInd; ++indProstog)
            if (kandidat % prosti[indProstog] == 0 ) {
                jeProst = FALSE;
                break;
            }
        if (jeProst)
            prosti[slobodanInd++] = kandidat;
        kandidat += 2;
    }

    /* kontrolni ispis - nije trazeno u zadatku */
    for (i = 0; i < MAX; ++i)
        printf("%d. %d\n", i+1, prosti[i]);

    return 0;
}
```

Varijabla `slobodanInd` ima vrijednost indeksa prvog slobodnog člana u polju prostih brojeva. Varijabla `kandidat` sadrži broj kojeg treba testirati da li je prost. Ta varijabla poprima vrijednosti 3, 5, 7, ...

Postupak uvećanja varijable `kandidat`, njenog testiranja i eventualnog ubacivanja u polje ponavlja se dok god se polje ne popuni s 500 prostih brojeva (tj. dok je `slobodanInd < MAX`). Testiranje kandidata se obavlja tako da se ispituje ostatak dijeljenja kandidata sa svakim prostim brojem koji se već nalazi u polju (osim `prosti[0]`). Kad se (ako se) naiđe na prosti broj s kojim je kandidat djeljiv, varijabla `jeProst` se postavlja na `FALSE`, a petlja se prekida. Ako je varijabla `jeProst` tijekom obavljanja for petlje ostala `TRUE`, tada se kandidat ubacuje u polje kao novi prosti broj.

Rješenje 6. zadatka

```
#include <stdio.h>
int main(void) {
    char niz[30+1]; /* vazno pitanje: zasto + 1? */
    int samog = 0, sug = 0, ostali = 0;
    int i = 0;
    gets(niz);

    while (niz[i] != '\0') {
        if (niz[i]=='A' || niz[i]=='E' || niz[i]=='I' ||
            niz[i]=='O' || niz[i]=='U' || niz[i]=='a' ||
            niz[i]=='e' || niz[i]=='i' ||
            niz[i]=='o' || niz[i]=='u')
            ++samog;
        else if (niz[i]>='A' && niz[i]<='Z' || niz[i]>='a' && niz[i]<='z')
            ++sug;
        else
            ++ostali;
        ++i;
    }
    printf("Samoglasnika: %d\n", samog);
    printf("Suglasnika: %d\n", sug);
    printf("Ostalih: %d\n", ostali);
    return 0;
}
```

Rješenje 7. zadatka

```
#include <stdio.h>
#define MAXNIZ 8

int main(void) {
    char niz[MAXNIZ+1];
    int ascii, i = 0;
    do {
        scanf("%d", &ascii);
        if (ascii >= 'A' && ascii <= 'Z') {
            niz[i] = ascii;
            ++i;
        }
    } while (i < MAXNIZ && ascii >= 'A' && ascii <= 'Z');

    /* zasto je sljedeća naredba važna? */
    niz[i] = '\0';

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

Rješenje 8. zadatka

```
#include <stdio.h>
#define MAXPOLJE 10

int main(void) {
    int ulaz[MAXPOLJE];
    int parni[MAXPOLJE];
    int neparni[MAXPOLJE];
    int n, i, brojParnih = 0, brojNeparnih = 0;

    do {
        printf("Unesite n: ");
        scanf("%d", &n);
    } while (n < 3 || n > 10);

    for (i = 0; i < n; ++i) {
        printf("Unesite %d. clan: ", i+1);
        scanf("%d", &ulaz[i]);
    }

    for (i = 0; i < n; ++i)
        if (ulaz[i] % 2 == 0)
            parni[brojParnih++] = ulaz[i];
        else
            neparni[brojNeparnih++] = ulaz[i];

    printf("\nPolje ulaz: ");
    for (i = 0; i < n; ++i)
        printf("%d ", ulaz[i]);

    printf("\nPolje parni: ");
    for (i = 0; i < brojParnih; ++i)
        printf("%d ", parni[i]);

    printf("\nPolje neparni: ");
    for (i = 0; i < brojNeparnih; ++i)
        printf("%d ", neparni[i]);

    printf("\n");

    return 0;
}
```


Rješenje 9. zadatka

```
#include <stdio.h>
#define MAXNIZ 100
int main (void) {
    char niz[MAXNIZ+1];
    int poz, i, dulj = 0;

    gets(niz);
    scanf("%d", &poz);
    /* koliko je niz dugacak */
    while (niz[dulj] != '\0')
        ++dulj;

    if (poz >= 1 && poz <= dulj) {
        /* Sve znakove od indeksa poz-1 do indeksa dulj
           posmaknuti za jedan u lijevo. Uocite, posmaknut ce se
           i \0, tako da ce niz na kraju biti kraci za jedan znak */
        for (i = poz - 1; i < dulj; ++i)
            niz[i] = niz[i+1];

        printf("%s\n", niz);
    }
    else {
        printf("Neispravna pozicija\n");
    }
    return 0;
}
```

Rješenje 10. zadatka

```
#include <stdio.h>
#define MAXNIZ 100
int main (void) {
    char niz[MAXNIZ+2]; /* Uociti +2 */
    int poz, i, dulj = 0;

    gets(niz);
    scanf("%d", &poz);
    /* koliko je niz dugacak */
    while (niz[dulj] != '\0')
        ++dulj;

    if (poz >= 1 && poz <= dulj) {
        /* Sve znakove od indeksa poz-1 do indeksa dulj
           posmaknuti za jedan u desno, ali mora se zapoceti
           od zadnjeg znaka. Uocite, posmaknut ce se
           i \0, tako da ce niz na kraju biti dulji za jedan znak */
        for (i = dulj; i >= poz - 1; --i)
            niz[i+1] = niz[i];

        /* na poziciju poz, odnosno indeks poz-1 upisi slovo X */
        niz[poz-1] = 'X';

        printf("%s\n", niz);
    }
    else {
        printf("Neispravna pozicija\n");
    }
    return 0;
}
```

Rješenje 11. zadatka

```
#include <stdio.h>
#define MAXNIZ 20
int main (void) {
    char niz1[MAXNIZ+MAXNIZ+1]; /* Uociti +MAXNIZ+1 */
    char niz2[MAXNIZ+1];
    int poz, i, dulj1 = 0, dulj2 = 0;

    gets(niz1);
    gets(niz2);
    scanf("%d", &poz);

    /* koliko je niz1 dugacak */
    while (niz1[dulj1] != '\0')
        ++dulj1;
    /* koliko je niz2 dugacak */
    while (niz2[dulj2] != '\0')
        ++dulj2;

    if (poz >= 1 && poz <= dulj1) {
        /* Sve znakove u niz1 od indeksa poz-1 do indeksa dulj1
           posmaknuti za dulj2 u desno, ali mora se zapoceti
           od zadnjeg znaka. Uocite, posmaknut ce se
           i \0, tako da ce niz1 na kraju biti dulji za dulj2 znakova */
        for (i = dulj1; i >= poz - 1; --i)
            niz1[i+dulj2] = niz1[i];
        /* od pozicije poz, odnosno indeksa poz-1 u niz1
           upisati sadrzaj niza niz2 */
        for (i = 0; i < dulj2; ++i)
            niz1[poz-1+i] = niz2[i];

        printf("%s\n", niz1);
    }
    else {
        printf("Neispravna pozicija\n");
    }
    return 0;
}
```

Rješenje 12. zadatka

```
#include <stdio.h>
#define MAXNIZ 20
int main (void) {
    char niz1[MAXNIZ+1];
    char niz2[MAXNIZ+1];
    char niz3[MAXNIZ+MAXNIZ+1]; /* Uociti +MAXNIZ+1 */
    int i, ind3; /* ind3 ce se koristiti za indeks u trecem nizu */

    gets(niz1);
    gets(niz2);

    /* prepisati niz1 u niz3 */
    i = ind3 = 0;
    while (niz1[i] != '\0')
        niz3[ind3++] = niz1[i++];

    /* dopisati niz2 na kraj niza niz3 */
    i = 0;
    while (niz2[i] != '\0')
        niz3[ind3++] = niz2[i++];

    /* na kraj niza niz3 upisati \0 */
    niz3[ind3] = '\0';

    printf("%s\n", niz3);

    return 0;
}
```

Rješenje 13. zadatka - varijanta u kojoj se frekvencije 'pamte' u polju

```
#include <stdio.h>
#define MAXNIZ 80
#define DG 'a'
#define GG 'z'
int main(void) {
    char niz[MAXNIZ+1], pozicija_u_nizu, c_iz_abecede;
    int brojac[GG-DG+1] = {0};
    gets(niz);

    pozicija_u_nizu = 0;
    while (niz[pozicija_u_nizu] != '\0') {
        brojac[niz[pozicija_u_nizu]-DG]++;
        ++pozicija_u_nizu;
    }
    for (c_iz_abecede = DG; c_iz_abecede <= GG; ++c_iz_abecede) {
        printf("slovo '%c' pojavilo se %2d puta\n",
            , c_iz_abecede
            , brojac[c_iz_abecede - DG]);
    }

    return 0;
}
```

Rješenje 13. zadatka - varijanta u kojoj se ne koristi polje za pohranu frekvencija

```
#include <stdio.h>

#define MAXNIZ 80
#define DG 'a'
#define GG 'z'

int main(void) {
    char niz[MAXNIZ+1], pozicija_u_nizu, c_iz_abecede;
    int brojac;
    gets(niz);

    for (c_iz_abecede = DG; c_iz_abecede <= GG; ++c_iz_abecede) {
        pozicija_u_nizu = 0;
        brojac = 0;
        while (niz[pozicija_u_nizu] != '\0') {
            if (niz[pozicija_u_nizu] == c_iz_abecede) {
                ++brojac;
            }
            ++pozicija_u_nizu;
        }
        printf("slovo '%c' pojavilo se %2d puta\n", c_iz_abecede, brojac);
    }

    return 0;
}
```