



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA PRIMENJENE RAČUNARSKE NAUKE

Osnovi programiranja i programskih jezika

Računarske vežbe – vežba 4

Zimski semestar 2025/2026.

Studijski program: Informacioni inženjering

Matrice Stringovi

Višedimenzionalni nizovi

- Opšta sintaksa:
 - `tip_podatka ime[dim1][dim2]...[dimN]`
- Primer matrice (dvodimenzionalnog niza):
 - `float tabela[5][12];`
- Pristup elementu niza:
 - `b = tabela[1][3];`
 - `tabela[4][10] = 15.33;`

Primer I

Nacrtati dijagram toka algoritma i napisati C program kojim se učitavaju dve kvadratne matrice **A** i **B**, dimenzija $n \times n$, i potom ispisuje njihova suma **A+B** i proizvod **A*B**.

```

#include<stdio.h>
#define MAX 10

int main() {
    int n, i, j, k;
    int A[MAX][MAX], B[MAX][MAX], C[MAX][MAX];

    printf("Unesite velicinu celobrojnih kvadratnih matrica: ");
    scanf("%d", &n);

    printf("Unos matrice A:\n");
    for (i=0; i<n ; i++)
        for (j=0; j<n ; j++) {
            printf("A[%d][%d]=", i+1, j+1);
            scanf("%d", &A[i][j]);
        }

    /* Unesi matricu B ... */
    /* U matricu C upisujemo zbir matrica A i B: C = A + B */
    for (i=0; i<n ; i++)
        for (j=0; j<n ; j++)
            C[i][j]=A[i][j]+B[i][j];
    printf("\nIspis matrice C=A+B:\n");
    for (i=0; i<n ; i++) {
        for (j=0; j<n ; j++)
            printf("%d\t", C[i][j]);
        printf("\n");
    }
    /* U matricu C upisujemo proizvod matrica A i B: C = A * B */
    for (i=0; i<n; i++)
        for (j=0; j<n; j++) {
            C[i][j]=0;
            for (k=0; k<n; k++)
                C[i][j]+=A[i][k]*B[k][j];
        }
    /* ... Ispis matrice C=A*B* */
    return 0;
} // Završi samostalno

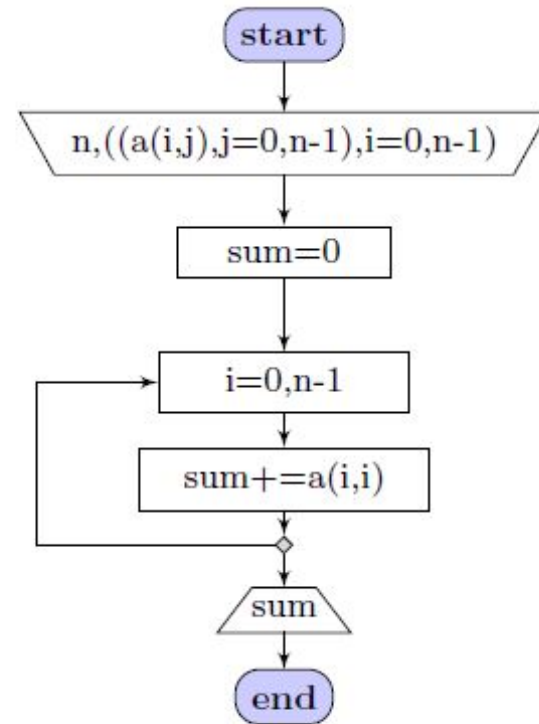
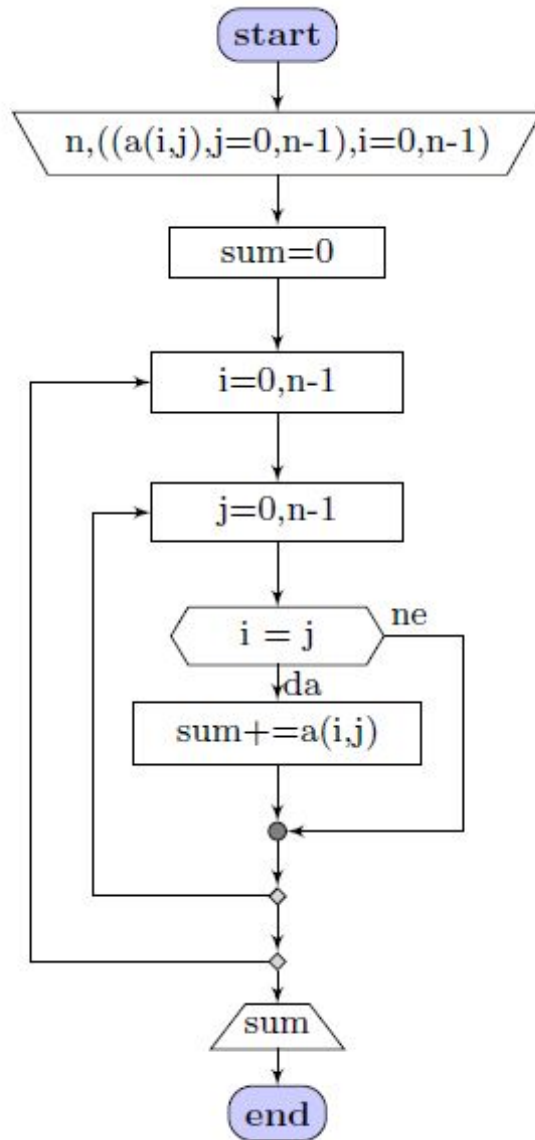
```

Zadatak I.

Nacrtati dijagram toka algoritma i napisati C program kojim se učitava kvadratna matrica **A**, dimenzija $n \times n$ ($n \geq 2$), i ispisuje suma elemenata koji se nalaze:

- a) u 2. vrsti,
- b) u 1. koloni,
- c) na glavnoj dijagonali,
- d) na sporednoj dijagonali,
- e) ispod glavne dijagonale,
- f) iznad sporedne dijagonale.

Zadatak I. – Rešenje pod c)



Zadatak 2.

Nacrtati dijagram toka algoritma i napisati C program kojim se učitava matrica **A** i razmenjuje:

- a) k -ta vrsta sa l -tom vrstom (celobrojne parametre k i l zadaje korisnik),
- b) vrsta koja sadrži element najveće vrednosti sa vrstom koja sadrži element najmanje vrednosti.

Prikazati matricu nakon transformacija.

Zadatak za domaći 1.

Nacrtati dijagram toka algoritma i napisati C program za transponovanje kvadratne matrice sa realnim elementima. Na početku programa potrebno je učitati dimenziju i elemente matrice. Prikazati matricu nakon transformacije (Zadatak 3.12 iz zbirke L. K.)

Zadatak za domaći 2.

Nacrtati dijagram toka algoritma i napisati C program za uređivanje kolona pravougaone matrice sa realnim elementima po neopadajućem redosledu zbrova elemenata po kolonama. Prikazati matricu nakon transformacije (Zadatak 3.13 iz zbirke L. K.)

Zadatak 3.

Nacrtati dijagram toka algoritma i napisati C program koji na šahovsku tablu (8x8) postavlja kraljicu na poziciju koju korisnik unosi (unos x i y koordinate). Na polju na kom je kraljica treba da se smesti broj 2, na poljima koje kraljica napada treba da se smesti broj 1 a na svim ostalim mestima treba da se smesti broj 0.

Zadatak za domaći 3.

Nacrtati dijagram toka algoritma i napisati C program koji na šahovsku tablu (8x8) postavlja kraljice tako da se ne napadaju međusobno, ali da zajednički napadaju sva ostala polja na tabli. Polja na kojima je postavljena kraljica treba obeležiti brojem 2, napadnuta polja brojem 1. Ispisati dobijenu matricu na standardni izlaz.

Zadatak 4.

Nacrtati dijagram toka algoritma i napisati C program koji modeluje ćelijski automat poznat kao [Langtonov mrav](#). Zamislite neograničen prostor podeljen na 2D mrežu ravnomernih kvadrata koji mogu biti beli ili crni. Za potrebe zadatka neka taj prostor bude matrica 1001×1001 u čijem je centru koordinatni početak. Na početku su svi crni. U centru ovog polja je Langtonov mrav usmeren nagore. U svakom koraku mrav skrene 90 stepeni ulevo ako je polje (na kojem stoji) crno i 90 stepeni udesno ako je polje belo. Zatim obrne boju polja (na kojem stoji) i pomeri se napred za jedan kvadrat. Za zadati broj koraka n ispisati poziciju na kojoj će se mrav naći nakon n koraka, a ako napusti tablu 1001×1001 ispisati da je izašao van prostora.

Sortiranje nizova

- Postoje različiti algoritmi za sortiranje:
 - sortiranje selekcijom (engl. *selection sort*)
 - sortiranje umetanjem (engl. *insertion sort*)
 - sortiranje zamenjivanjem (engl. *bubble sort*)
 - sortiranje spajanjem (engl. *merge sort*)
 - brzo sortiranje (engl. *quick sort*)
 - sortiranje primenom gomile (engl. *heapsort*)
 - ...

Selection sort (primer 2)

```
#include <stdio.h>

int main() {
    int niz[100], n, i, j, minIndex, temp;
    printf("Unesite broj elemenata:\n");
    scanf("%d", &n);
    printf("Unesite elemente:\n");
    for (i = 0; i < n; i++) {
        printf("niz[i] = ");
        scanf("%d", &niz[i]);
    }
    for (i = 0; i < n - 1; i++) {           //proci cemo kroz sve elemente
        minIndex = i;                       //pretpostavljamo da je trenutni element najmanji
        for (j = i + 1; j < n; j++) {       //pronalazimo zaista najmanji element u ostatku niza
            if (niz[minIndex] > niz[j]) minIndex = j;
        }
        if (minIndex != i) { //menjamo mesta trenutnom i najmanjem elementu - osim ako je to isti element
            temp = niz[i];
            niz[i] = niz[minIndex];
            niz[minIndex] = temp;
        }
    }
    printf("Sortirani niz:\n");
    for (i = 0; i < n; i++) {
        printf("%d\n", niz[i]);
    }
    return 0;
}
```

String

- String je niz karaktera koji se završava **NULL** karakterom - `'\0'`
- Format specifikator za string u `printf` i `scanf` funkcijama je `%s`
- String konstante se pišu pod navodnicima:
 - `char s[] = "neki tekst";`
- Stringu nije moguće dodeliti vrednost string konstante osim prilikom definicije
 - `char s[100];`
 - `s = "neki string"; // greška, s je niz!`

Primer 3

```
#include <stdio.h>
#define MAX_LEN 101

int main() {
    char str1[MAX_LEN], str2[MAX_LEN];

    printf("Unesite string 1:");
    scanf("%s", str1);

    __fpurge(stdin);

    printf("Unesite string 2:");
    gets(str2);

    printf("\n%s\n", str1);
    printf("\n%s\n", str2);

    puts(str1);
    puts(str2);

    return 0;
}
```

Zadatak 5.

Nacrtati dijagram toka algoritma i napisati C program kojim se učitava string. Potom se učitani string ispisuje u učitanoj i obrnutom redosledu, zajedno sa informacijom o broju karaktera u stringu.

Zadatak 6.

Nacrtati dijagram toka algoritma i napisati C program kojim se ispisuje:

- a) prvih **k** znakova string promenljive **s**;
- b) poslednjih **k** znakova string promenljive **s**;
- c) deo stringa **s** od pozicije **p** na dužini **k**.

Zadatak za domaći 4.

Nacrtati dijagram toka algoritma i napisati C program koji proverava da li je učitani string palindrom. Program napraviti tako da radi za reči, a potom ga proširiti da radi i sa rečenicama (ignorišući znakove razmaka) kako bi string "Ana voli Milovana" program registrovao kao palindrom.

Zadatak 7.

Nacrtati dijagram toka algoritma i napisati C program koji kao ulaz prihvata jedan string i karakter, a određuje broj pojavljivanja tog karaktera u stringu.

Primer: `str="tatatatira"`, `c='a'`, izlaz je 4

Zadatak za domaći 5.

Nacrtati dijagram toka algoritma i napisati C program koji proverava da li su stringovi s_1 i s_2 jednaki.

Zadatak za domaći 6.

Nacrtati dijagram toka algoritma i napisati C program koji za string koji korisnik unese sa standardnog ulaza prebroji i prikaže broj:

- a) Svih slova
- b) Samoglasnika
- c) Suglasnika

Primer: `str="Danas je 29. oktobar."`, slova ima 14, suglasnika 8, samoglasnika 6.

Zaglavlje `string.h`

Korišćenje funkcija iz zaglavlja `string.h` olakšava rad sa stringovima.

Neke od najčešće korišćenih funkcije su:

- `strlen` – dužina stringa
- `strcpy` – kopiranje stringa
- `strcmp` – poređenje stringova
- `strcat` – konkatencija (spajanje) stringova
- `strstr` – broj pojavljivanja jednog stringa u drugom

Primer 4

Sledeći primer ilustruje korišćenje funkcije `strcat()`:

```
#include <stdio.h>
#include <string.h>
#define MAX_LEN 31

int main() {
    char s1[MAX_LEN] = "Veni, ";
    char s2[] = "vidi, ";

    printf("%s\n", strcat(s1, s2));
    strcat(s1, "vici.");
    printf("%s\n", s1);

    return 0;
}
```

Primer 5

Nacrtati dijagram toka algoritma i napisati C program za učitavanje stringova koje se prekida kada se naiđe na **EOF** (tada `gets()` vraća **NULL**), prazan string ili se dostigne predviđen broj članova niza koji se zadaje konstantom **MAX**. Obezbediti ispis unetog niza.

Primer 5

```
#include <stdio.h>
#include <string.h>
#define DUZINA 31
#define MAX 50
#define STOP ""

int main() {
    char str[MAX][DUZINA];
    int i=0, j;

    while (gets(str[i]) != NULL &&
           strcmp(str[i], STOP) !=0 &&
           i++ < MAX);

    for (j=0; j<i; j++) {
        puts(str[j]);
    }

    return 0;
}
```

Primer 5

```
#include <stdio.h>
#include <string.h>
#define DUZINA 31
#define MAX 50
#define STOP ""

int main() {
    char str[MAX][DUZINA];
    int i=0, j;

    while (i<MAX) {
        if (fgets(str[i], DUZINA, stdin) == NULL) break;
        size_t len = strlen(str[i]);
        if (len > 0 && str[i][len - 1] == '\n') str[i][len - 1] = '\0';
        if (strcmp(str[i], STOP) == 0) break;
        i++;
    }

    for (j=0; j<i; j++) {
        puts(str[j]);
    }

    return 0;
}
```

Primer 6

Sledeći primer ilustruje korišćenje funkcije `strcpy()`:

```
#include <stdio.h>
#include <string.h>
#define DUZINA 31

int main() {
    char str[DUZINA];

    strcpy(str, "Znanje");
    puts(str);
    puts(strcpy(str, "Imanje"));

    return 0;
}
```

Zadatak 8.

Nacrtati dijagram toka algoritma i napisati C program koji u unesenom stringu *s* mala slova pretvara u velika, i obrnuto. Ostale karaktere u stringu (cifre, znakove interpunkcije, itd.) program ne sme modifikovati.

Zadatak 9.

Nacrtati dijagram toka algoritma i napisati C program koji uneseni string s pretvara u realan broj. Realan broj se definiše kao niz od jedne ili više cifara, nakon koje može biti tačka, nakon koje može biti još nula ili više cifara. Ukoliko je prilikom konverzije došlo do greške (tj. ako string sadrži nedozvoljeni karakter), ispisati odgovarajuću poruku.

Zadatak 10.

Nacrtati dijagram toka algoritma i napisati C program koji od korisnika prihvata niz stringova i ispisuje ih na ekran. Program treba od korisnika najpre da ućita dućzinu niza (odnosno broj stringova), a zatim svaki pojedinaćni string.

Zadatak I I.

Nacrtati dijagram toka algoritma i napisati C program koji u datom stringu određuje:

- a) broj reči;
- b) broj reči koje počinju sa M;
- c) broj reči čiji su prvi i poslednji znak jednaki;
- d) dužinu najkraće reči.

Zadatak za domaći 7.

Nacrtati dijagram toka algoritma i napisati C program koji unešeni string šifruje Cezarovom šifrom za KLJUČ (ceo broj) mesta udesno i ispisuje šifrovan string. Program treba da radi samo za **mala slova abecede**. Rezultujuće slovo se računa kao slovo na KLJUČ mesta udesno od početnog. Primeri kako promeniti pojedinačno slovo za ključ 5:

1. A prelazi u F (A > ~~BCDEF~~)
2. J prelazi u O
3. W prelazi u B (W > ~~XYZAB~~)
4. Y prelazi u D

Zadatak za domaći 7.1.

Izmeniti zadatak 17 tako da se ponaša kao dešifратор.
(menja slova ulevo)

Zadatak za domaći 8.

Nacrtati dijagram toka algoritma i napisati C program koji za uneseni string proverava da li je moguće da taj string bude palindrom. Ako je moguće, treba da se ispiše potencijalni palindrom, ako ne, treba ispisati "-1". Na ulazu su dozvoljena samo mala slova abecede i tačka. Tačka je džoker znak, tj. menja bilo koje slovo.

Primeri:

```
ulaz="a.ba"  
izlaz="abba"
```

```
ulaz="a..b"  
izlaz=-1
```

Zadatak za domaći 9.

Nacrtati dijagram toka algoritma i napisati C program koji proverava evidenciju prolaska zmija na ulazu na festival. Sova stoji na ulazu i ukoliko vidi na ulazu zmijinu glavu piše slovo 'G', ako ugleda rep piše slovo 'R' a u ostalim trenucima kada evidentira piše karakter '.' ukoliko zmije nema ili vidi da još uvek prolazi telom kroz ulaz. Evidencija se unosi kao string, a na izlazu se ispisuje OK ako je sve u redu, tj. JOK ako nije. Pošto zmije pristižu redom, u evidenciji se ne sme desiti da rep dođe pre glave, kao ni više glava ili repova zaredom, kao ni glava nakon koje ne dolazi rep.

Primeri:

Ulaz=...G.....RGR.G..R.GR....G....R..	
Izlaz=OK	
Ulaz=..G.....GRR.	Ulaz=...RGR
Izlaz=JOK	Izlaz=JOK

Zadatak za domaći 10.

Nacrtati dijagram toka algoritma i napisati C program koji upoređuje 2 unešena stringa i računa najveću i najmanju razliku između njih i ispisuje ih. Oba stringa su iste dužine i sastoje se samo iz malih slova i upitnika koji je džoker znak i može da zameni bilo koje slovo.

Primeri:

Ulaz:

a?c

??b

Izlaz:

1 3

Ulaz:

?abac

aba?w

Izlaz:

3 5