

Programiranje - programski jezik C

Davorka Jandrlić, Andrija Jandrlić, Goran Lazović

1 Blok I

1.1 Linearna struktura programa

Zadatak 1.1 *Napisati program koji na standardnom izlazu ispisuje 'Zdravo svete!'.*

```
#include <stdio.h>

int main() {
    printf("Zdravo svete!\n");
    system("pause");
    return 0;
}
```

Zadatak 1.2 Šta se ispisuje sledećim programom?

```
#include <stdio.h>

int main() {
    printf("\nZdravo, svima\n\n");
    printf("\n\tprelazak u novi red\n");
    printf("\t\ttabulator\n");
    printf("\\\\\\tkosa crta\n");
    printf("%%%\tprocenat\n");

    system("pause");
    return 0;
}
```

Zadatak 1.3 Napisati program koji ilustruje deklaraciju celobrojne promenljive i ispis njene vrednosti.

```
#include <stdio.h>

int main() {
    int i = 2;
    printf("Vrednost promenjive i je %d\n", i);
```

```

    system("pause");
    return 0;
}

```

Zadatak 1.4 *Napisati program u kojem sa standardnog ulaza unosi vrednost celobrojne promenljive i ispisuje na standardni izlaz.*

```

#include <stdio.h>

int main() {
    int x;
    printf("Unesite ceo broj: ");
    /* Obratiti paznju na znak & pre imena promenljive
       u funkciji scanf. */
    scanf("%d",&x);

    /* U funkciji printf ne treba stavljati &. */
    printf("Uneli ste broj %d\n", x);

    system("pause");
    return 0;
}

```

Zadatak 1.5 *Napisati program u kojem sa standardnog ulaza unosi vrednost realne promenljive i ispisuje zaokružen na dve decimale na standardni izlaz.*

```

#include <stdio.h>

int main() {
    float x;
    printf("Unesite realan broj: ");
    scanf("%f", &x);
    printf("Broj zaokruzen na dve decimale je: %.2f\n", x);

    system("pause");
    return 0;
}

```

Zadatak 1.6 *Napisati program koji sabira dva cela broja zadata sa standardnog ulaza.*

```

#include <stdio.h>

int main() {
    int a, b, c;
    printf("Unesi prvi broj : ");

```

```

scanf("%d", &a);
printf("Unesi drugi broj : ");
scanf("%d", &b);
c = a + b;
printf("%d + %d = %d\n", a, b, c);

system("pause");
return 0;
}

```

Zadatak 1.7 *Napisati program kojim se razmenjuju vrednosti dva cela broja uneta sa standardnog ulaza.*

```

#include <stdio.h>

int main() {
    int a, b;
    int pom;
    printf("Unesi prvi broj : ");
    scanf("%d", &a);
    printf("Unesi drugi broj : ");
    scanf("%d", &b);

    printf("Pre razmene: a = %d, b = %d\n", a, b);

    pom = a;
    a = b;
    b = pom;

    printf("Posle razmene: a = %d, b = %d\n", a, b);

    system("pause");
    return 0;
}

```

Zadatak 1.8 *Napisati program kojim se demonstriraju aritmetičke operacije nad dva cela broja uneta sa standardnog ulaza.*

```

#include <stdio.h>

int main() {
    int a, b;
    printf("Unesi prvi broj: ");
    scanf("%d",&a);

    printf("Unesi drugi broj: ");
    scanf("%d",&b);

```

```

printf("%d + %d = %d\n", a, b, a + b);
printf("%d - %d = %d\n", a, b, a - b);
printf("%d * %d = %d\n", a, b, a * b);
printf("Celobrojni kolicnik a / b je: %d\n", a/b);
printf("Pogresan pokusaj racunanja realnog kolicnika a/b \
      je: %f\n", a/b);
printf("Realni kolicnik a/b je: %f\n", (float)a/(float)b);
printf("Ostatak pri deljenju a/b je: %d\n", a%b);

system("pause");
return 0;
}

```

Zadatak 1.9 *Napisati program koji izračunava obim i površinu kruga poluprečnika r, koji se unosi sa standardnog ulaza.*

```

#include <stdio.h>

#define PI 3.14159

int main() {
    int r;

    printf("Unesi poluprecnik kruga: ");
    scanf("%d",&r);

    printf("Obim: %g\n", 2 * r * PI);
    printf("Povrsina: %g\n", r * r * PI);

    system("pause");
    return 0;
}

```

Zadatak 1.10 *Šta se ispisuje sledećim programom?*

```

/*
Operator ++ uvecava vrednost promenjive za 1.
Operator -- je potpuno analogan operatoru ++ osim sto
      umanjuje vrednost za 1
*/
#include <stdio.h>
int main() {
    int x, y;
    int a = 0, b = 0;

    printf("Na pocetku : \na = %d\nb = %d\n", a, b);

```

```

    /* Ukoliko se vrednost izraza ne koristi, prefiksni i
    postfixni operator se ne razlikuju */
    a++;
    ++b;
    printf("Posle : a++; ++b; \na = %d\nb = %d\n", a, b);

    /* Prefiksni operator uvecava promenjivu, i rezultat
    je uvecana vrednost */
    x = ++a;

    /* Postfiksni operator uvecava promenjivu, i rezultat je
    stara (neuvecana) vrednost */
    y = b++;

    printf("Posle : x = ++a; \na = %d\nx = %d\n", a, x);
    printf("Posle : y = b++; \nb = %d\ny = %d\n", b, y);

    system("pause");
    return 0;
}

```

Domaći zadatak 1.1 *Napisati program za izračunavanje površine i zapremine pravog valjka zadatog svojom visinom H i poluprečnikom osnove r .*

Domaći zadatak 1.2 *Napisati program koji izračunava vrednost celog broja nakon njegovog umanjeanja od 10%.*

2 Blok II

2.1 Razgranata struktura programa

Zadatak 2.1 *Napisati program koji izračunava maksimum dva cela broja uneta sa ulaza.*

```

#include <stdio.h>

int main() {
    int a, b;
    int max; // Vrednost maksimuma
    printf("Unesi prvi broj: ");
    scanf("%d",&a);

    printf("Unesi drugi broj: ");
    scanf("%d",&b);
}

```

```

// Pretpostavimo da je prvi broj veci.
max = a;

// Ukoliko to nije slucaj, vrednost se azurira.
if(max < b) {
    max = b;
}

printf("Maksimum je: %d\n", max);

system("pause");
return 0;
}

```

Zadatak 2.2 *Napisati program koji za uneto x izračunava y po formuli:*

$$y = \begin{cases} \frac{-x^3}{3} & \text{za } x > 0 \\ \frac{x+1}{2} & \text{inače} \end{cases}$$

```

#include <stdio.h>
#include <math.h>

int main() {
    float x;
    float y;

    printf("x = ?\n");
    scanf("%f", &x);

    if(x > 0) {
        y = - pow(x, 3) / 3.0;
    }
    else {
        y = (x + 1) / 2;
    }

    printf("y = %f\n", y);

    system("pause");
    return 0;
}

```

Zadatak 2.3 *Napisati program koji proverava da li se od zadatih odsečaka a, b i c može konstruisati trougao. Ako može izračunati površinu trougla, ako ne ispisati odgovarajuću poruku.*

```

#include <stdio.h>

```

```

#include <math.h>

int main() {
    double a, b, c;

    printf("Unesite stranice trougla a = ?, b = ?, c = ?\n");
    scanf("%lf%lf%lf", &a, &b, &c);

    if(a + b > c && a + c > b && b + c > a) {
        double s = (a + b + c) / 2;
        double P = sqrt(s * (s - a) * (s - b) * (s - c));

        printf("Povrsina trougla je: %.2g\n", P);
    }
    else {
        printf("Od unetih stranica ne moze se formirati trougao!\n");
    }

    system("pause");
    return 0;
}

```

Zadatak 2.4 *Napisati program koji pronalazi realna rešenja kvadratne jednačine zadate u obliku*

$$ax^2 + bx + c = 0$$

```

#include <stdio.h>
#include <math.h>

int main() {
    float a, b, c;
    float D; // diskriminanta

    printf("Unesite koeficijente kvadratne jednacine \
          a = ?, b = ?, c = ?\n");
    scanf("%f%f%f", &a, &b, &c);

    if(a == 0.0) {
        printf("Greska a = 0, jednacina nije kvadratna!\n");

        system("pause");
        return 1;
    }

    // Izracunava se diskriminanta

```

```

D = b * b - 4 * a * c;
if(D == 0.0) {
    printf("Jednacina ima jedinstveno resenje \
          x1 = x2 = %.2f\n", - b / (2 * a));
}
else if(D > 0.0) {
    printf("Jednacina ima dva realna resenja:\n");
    printf("x1 = %.2f\n", (-b - sqrt(D)) / (2 * a));
    printf("x2 = %.2f\n", (-b + sqrt(D)) / (2 * a));
}
else {
    printf("Jednacina nema realnih resenja.\n");
}

system("pause");
return 0;
}

```

Zadatak 2.5 *Napisati program koji za zadati redni broj dana u nedelji ispisuje njegovo ime. U slučaju nekorektno unetog dana ispisati odgovarajuću poruku.*

```

#include <stdio.h>

int main() {
    // Redni broj dana u nedelji.
    int dan;

    printf("Redni broj dana u nedelji:\n");
    scanf("%d", &dan);

    switch(dan) {
        case 1:
            printf("Ponedeljak\n");
            break;
        case 2:
            printf("Utorak\n");
            break;
        case 3:
            printf("Sreda\n");
            break;
        case 4:
            printf("Cetvrtak\n");
            break;
        case 5:
            printf("Petak\n");
            break;
    }
}

```



```

        case 6:
            printf("Subota\n");
            break;
        case 7:
            printf("Nedelja\n");
            break;
        default:
            printf("Nekorektan unos, redni broj dana treba da \
                je u intervalu [1, 7]\n");
    }

    system("pause");
    return 0;
}

```

Zadatak 2.6 *Napisati program kojim se simulira rad uproštenog kalkulatora. Sa ulaza se unosi ceo broj, zatim jedna od aritmetickih operacija +, -, *, /, %, i na kraju drugi ceo broj, a zatim se na standardni izlaz ispisuje vrednost zadatog izraza.*

```

#include <stdio.h>

int main() {
    int a, b; // operandi
    char op;  // operacija

    printf("Unesti izraz u obliku operand operacija operand:\n");
    scanf("%d%c%d", &a, &op, &b);

    switch(op) {
        case '+':
            printf("%d %c %d = %d\n", a, op, b, a + b);
            break;
        case '-':
            printf("%d %c %d = %d\n", a, op, b, a - b);
            break;
        case '*':
            printf("%d %c %d = %d\n", a, op, b, a * b);
            break;
        case '/':
            // Obratiti paznju na deljenje!
            printf("%d %c %d = %f\n", a, op, b, a / (b + 0.0));
            break;
        case '%':
            printf("%d %c %d = %d\n", a, op, b, a % b);
            break;
    }
}

```

```

        default:
            // Obratiti paznju na ispis \ i %
            printf("Nekorektan unos, operacija mora biti jedna \
                od: +, -, *, \\\, %%\n");
    }

    system("pause");
    return 0;
}

```

Domaći zadatak 2.1 *Napisati program koji izračunava maksimum i minimum tri cela broja sa ulaza.*

Domaći zadatak 2.2 *Napisati program koji rešava linearnu jednačinu*

$$ax + b = 0$$

.

Domaći zadatak 2.3 *Ispitati da li se prave*

$$y = k_1x + n_1$$

i

$$y = k_2x + n_2$$

seku i ako se seku naći koordinate presečne tačke.

Domaći zadatak 2.4 *Napisati program koji za zadati redni broj meseca ispisuje broj dana u tom mesecu. Za mesec februar pretpostaviti da ima 28 dana.*

3 Blok III

3.1 Ciklusi

Zadatak 3.1 *Napisati program koji štampa prvih 10 prirodnih brojeva.*

```

#include <stdio.h>

int main() {
    int i; // Brojac u petlji

    for(i = 1; i <= 10; i++) {
        printf("%d\n", i);
    }
}

```

```

        system("pause");
        return 0;
    }

```

Zadatak 3.2 *Napisati program koji izračunava $n!$.*

```

#include <stdio.h>

int main() {
    int n;        // Broj ciji se faktoriyel racuna
    long fakt;    // Vrednost faktoriyela
    int i;        // Brojac u petljama

    printf("n = ?\n");
    scanf("%d", &n);

    // Inicijalizacija (obratiti paznju!)
    fakt = 1;
    for(i = 1; i <= n; i++) {
        fakt *= i;
    }

    printf("%d! = %ld\n", n, fakt);

    system("pause");
    return 0;
}

```

Zadatak 3.3 *Šta se dobija izvršavanjem sledećeg dela programa.*

```

int i, j;
for(i = 1; i <= 3; i++) {
    for(j = 1; <= 3; j++) {
        printf("%d * %d = %d\t", i, j, i * j);
    }
    printf("\n");
}

```

Zadatak 3.4 *Napisati program koji prepisuje karaktere unete sa standardnog ulaza na standardni izlaz.*

```

#include <stdio.h>

int main() {
    char c;

    while((c = getchar()) != EOF) {

```

```

        putchar(c);
    }

    system("pause");
    return 0;
}

```

Zadatak 3.5 *Napisati program koji vrši prebrojavanje cifara u tekstu unetog sa ulaza.*

```

#include <stdio.h>

int main() {
    char c;
    int br_cifara = 0;

    while((c = getchar()) != EOF) {
        if(c >= '0' && c <= '9') {
            br_cifara++;
        }
    }

    printf("Broj unetih cifara je: %d\n", br_cifara);

    system("pause");
    return 0;
}

```

Napomena: Zadatak je mogao da se reši korišćenjem funkcije `isdigit(c)` iz zaglavlja `ctype.h`.

Zadatak 3.6 *Napisati program koji izračunava sumu pozitivnih celih brojeva koji se zadaju sa standardnog ulaza. Unos prekinuti kada se unese 0.*

```

#include <stdio.h>

int main() {
    int suma;

    suma = 0;
    // Beskonacna petlja
    while(1) {
        int x;

        printf("Broj (0 za kraj): ");
        scanf("%d", &x);
    }
}

```

```

        if(x == 0) {
            break;
        }

        if(x > 0) {
            suma += x;
        }
    }

    printf("Suma pozitivnih je %d\n", suma);

    system("pause");
    return 0;
}

```

Zadatak 3.7 *Počevši treniranje takmičar je prve nedelje pretrčao A kilometara, a kasnije svake nedelje je razdaljinu povećavao za 10%. Odrediti nedelju u kojoj je pretrčao rastojanje jednako dužini maratona (rastojanje 42.195km).*

```

#include <stdio.h>

#define DUZINA_MARATONA 42.195

int main() {
    float A;
    int nedelja;

    printf("Unesite duzinu kojom je takmicar poceo treninge, u km: ");
    scanf("%f", &A);

    nedelja = 1;
    while(A < DUZINA_MARATONA) {
        A *= 1.1;
        nedelja++;
    }

    printf("U %d nedelji takmicar je pretrcao %.3f km\n", nedelja, A);

    system("pause");
    return 0;
}

```

Domaći zadatak 3.1 *Napisati program kojim se izračunava suma cifara celog broja unetog sa standardnog ulaza.*

Domaći zadatak 3.2 *Napisati program kojim se obrću cifre celog broja unetog sa standardnog ulaza.*

Domaći zadatak 3.3 *Napisati program kojim se izračunava sledeća suma:*

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

.

4 Blok IV

4.1 Nizovi

Zadatak 4.1 *Šta se ispisuje sledećim programom:*

```
#include <stdio.h>

int main() {
    int a[10];
    int i;

    for (i = 0; i<10; i++){
        printf("a[%d]=", i);
        scanf("%d", &a[i]);
    }

    printf("Unazad: \n");

    for (i = 9; i >= 0; i--) {
        printf("a[%d]=%d\n", i, a[i]);
    }

    system("pause");
    return 0;
}
```

Zadatak 4.2 *Šta se ispisuje sledećim programom:*

```
#include <stdio.h>

int main() {
    /* Niz se inicijalizuje tako sto mu se navode vrednosti
       u viticastim 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);

/* Ispisuju se nizovi */
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]);
}

system("pause");
return 0;
}

```

Zadatak 4.3 *Napisati program koji izračunava sumu elemenata i srednju vrednost niza celih brojeva dužine ne veće od 50.*

```

#include <stdio.h>

#define MAX_DUZINA_NIZA 50

int main() {
    int a[MAX_DUZINA_NIZA];
    int n; // Stvarna duzina niza
    int suma;

    int i;

    printf("Duzina niza? (<= %d): ", MAX_DUZINA_NIZA);
    scanf("%d", &n);

    suma = 0;

    printf("Unesite elemente niza:\n");
    for(i = 0; i < n; i++) {
        printf("a[%d] = ?\n", i);
        scanf("%d", &a[i]);
        // Suma se izracunava odmah nakon unosa tekuceg elementa
    }
}

```

```

        suma += a[i];
    }

    printf("Suma = %d, srednja vrednost = %f\n", \
           suma, suma / (n + 0.0));

    system("pause");
    return 0;
}

```

Zadatak 4.4 *Napisati program koji pronalazi minimalni element niza realnih brojeva dužine ne veće od 50, i njegov indeks.*

```

#include <stdio.h>

#define MAX_DUZINA_NIZA 50

int main() {
    float a[MAX_DUZINA_NIZA];
    int n;          // Stvarna duzina niza
    int min_ix;     // Indeks minimalnog elementa niza

    int i;

    printf("Duzina niza? (<= %d): ", MAX_DUZINA_NIZA);
    scanf("%d", &n);

    // Pretpostavimo da je prvi element niza najmanji
    min_ix = 0;

    printf("Unesite elemente niza:\n");
    for(i = 0; i < n; i++) {
        printf("a[%d] = ?\n", i);
        scanf("%f", &a[i]);

        /*
           Uporedjuje se tekuci element sa elementom na poziciji
           min_ix, ukoliko je tekuci element manji azurira se
           vrednost indeksa najmanjeg elementa.
        */
        if(a[i] < a[min_ix]) {
            min_ix = i;
        }
    }

    printf("Min = %f na poziciji = %d\n", \

```



```

        a[min_ix], min_ix);

    system("pause");
    return 0;
}

```

Zadatak 4.5 *Dat je niz celih brojeva dužine ne veće od 50, i ceo broj x. Napisati program koji proverava koliko se puta zadati broj pojavljuje u nizu.*

```

#include <stdio.h>

#define MAX_DUZINA_NIZA 50

int main() {
    int a[MAX_DUZINA_NIZA];
    int n; // Stvarna duzina niza

    int x; // Broj koji se trazi
    int br_pojavljivanja;

    int i;

    printf("Duzina niza? (<= %d): ", MAX_DUZINA_NIZA);
    scanf("%d", &n);

    printf("Unesite elemente niza:\n");
    for(i = 0; i < n; i++) {
        printf("a[%d] = ?\n", i);
        scanf("%d", &a[i]);
    }

    printf("Unesite broj koji se trazi:\n");
    scanf("%d", &x);

    br_pojavljivanja = 0;
    for(i = 0; i < n; i++) {
        if(a[i] == x) {
            br_pojavljivanja++;
        }
    }

    printf("Broj %d, pojavljuje se %d puta\n", \
        x, br_pojavljivanja);

    system("pause");
    return 0;
}

```

Zadatak 4.6 *Napisati program koji računa skalarni proizvod dva vektora u n - dimenzionom vektorskom prostoru ($n \leq 50$).*

```
#include <stdio.h>

#define MAX_DUZINA_NIZA 50

int main() {
    float a[MAX_DUZINA_NIZA];
    float b[MAX_DUZINA_NIZA];
    int n; // Stvarna duzina niza

    float skalarni_proizvod;

    int i;

    printf("Duzina niza? (<= %d): ", MAX_DUZINA_NIZA);
    scanf("%d", &n);

    printf("Unesite prvi vektor:\n");
    for(i = 0; i < n; i++) {
        printf("a[%d] = ?\n", i);
        scanf("%f", &a[i]);
    }

    printf("Unesite drugi vektor:\n");
    for(i = 0; i < n; i++) {
        printf("b[%d] = ?\n", i);
        scanf("%f", &b[i]);
    }

    skalarni_proizvod = 0.0;
    for(i = 0; i < n; i++) {
        skalarni_proizvod += a[i] * b[i];
    }

    printf("Skalarni proizvod je %f\n", \
           skalarni_proizvod);

    system("pause");
    return 0;
}
```

Zadatak 4.7 *Napisati program koji sortira rastuće niz celih brojeva dužine n , $n \leq 50$. (selection sort)*

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

```

#define MAX_DUZINA_NIZA 50

int main() {
    int a[MAX_DUZINA_NIZA];
    int n;  // Stvarna duzina niza

    int i, j;

    printf("Duzina niza? (<= %d): ", MAX_DUZINA_NIZA);
    scanf("%d", &n);

    printf("Unesite elemente niza:\n");
    for(i = 0; i < n; i++) {
        printf("a[%d] = ?\n", i);
        scanf("%d", &a[i]);
    }

    printf("Pre sortiranja:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", a[i]);
    }
    printf("\n");

    for(i = 0; i < n - 1; i++) {
        for(j = i + 1; j < n; j++) {
            if(a[i] > a[j]) {
                int pom = a[i];
                a[i] = a[j];
                a[j] = pom;
            }
        }
    }

    printf("Posle sortiranja:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", a[i]);
    }
    printf("\n");

    system("pause");
    return 0;
}

```

Zadatak 4.8 *Rešiti prethodni zadatak primenom bubble sort algoritma.*

```

for(i = n - 1; i > 0; i--) {

```

```

        for(j = 0; j < i; j++) {
            if(a[j] > a[j + 1]) {
                int pom = a[j];
                a[j] = a[j + 1];
                a[j + 1] = pom;
            }
        }
    }
}

```

Domaći zadatak 4.1 *Napisati program koji za niz celih brojeva dužine $n \leq 20$ proverava da li je palindrom. Niz je palindrom ako je redosled elemeneta isti kada se niz posmatra sprede i odpozadi.*

Domaći zadatak 4.2 *Na Svetskom prvenstvu u trčanju za muškarce na 100m, nizom dužine $n \leq 10$ su predstavljeni rezultati. Prikazati rezultate sortirane rastuće po vremenu trčanja. Ako je poznato da svetski rekord iznosi 9.58sec, ispitati da li je na ovom takmičenju postavljen novi.*

Domaći zadatak 4.3 *Napisati program koji izračunava vrednost polinoma stepena ne većeg od 20 čiji su koeficijenti realni brojevi zadati nizom, u nekoj tački x .*

5 Blok V

5.1 Nizovi, niske i funkcije

Zadatak 5.1 *Funkcije i nizovi:*

```

#include <stdio.h>

/* Funkcija koja stampa niz celih brojeva */
void print_array(int x[], int n) {
    int i;
    for(i = 0; i < n; i++) {
        printf("%d ", x[i]);
    }

    printf("\n");
}

/* Funkcija koja stampa niz realnih brojeva */
void print_arrayd(double x[], int n) {
    int i;
    for(i = 0; i < n; i++) {
        printf("%f ", x[i]);
    }
}

```

```

        printf("\n");
    }

    /* Funkcija koja siftuje elemente niza ulevo */
    void cycle_left(int x[], int n) {
        int i;

        int first = x[0];
        for(i = 1; i < n; i++) {
            x[i - 1] = x[i];
        }
        x[n - 1] = first;
    }

    /* Funkcija koja siftuje elemente niza udesno */
    void cycle_right(int x[], int n) {
        int i;

        int last = x[n - 1];
        for(i = n - 2; i >= 0; i--) {
            x[i + 1] = x[i];
        }
        x[0] = last;
    }

    /* Funkcija koja obrce elemente niza */
    void reverse_array(int x[], int n) {
        int i;

        for(i = 0; i < n / 2; i++) {
            int pom = x[i];
            x[i] = x[n - i - 1];
            x[n - i - 1] = pom;
        }
    }

    /* Funkcija koja sortira elemente niza rastuce */
    void bubble_sort(int x[], int n) {
        int i, j;

        for(j = n - 1; j > 1; j--) {
            for(i = 0; i < j; i++) {
                if(x[i] < x[i + 1]) {
                    int pom = x[i];
                    x[i] = x[i + 1];
                    x[i + 1] = pom;
                }
            }
        }
    }

```

```

        x[i + 1] = pom;
    }
}
}

/* Funkcija koja transformise elemente niza, demonstracija poziva funkcije unutar funkcije */
void transform(int x[], int n) {
    int i;
    double y[n];

    for(i = 0; i < n / 2; i++) {
        y[i] = (x[2 * i] + x[2 * i + 1]) / 2.0;
    }

    print_arrayd(y, n / 2);
}

/* Glavni program i demonstracija poziva funkcija. Napomena:
 * Sve navedene funkcije su tipa void - nemaju povratnu vrednost */
int main() {
    int x[] = {-2, 4, 0, -1, -6, 8, 12, 6, 7};

    int n = sizeof(x)/sizeof(int);

    printf("Niz inicijalno:\n");
    print_array(x, n);

    printf("Bubble:\n");
    bubble_sort(x, n);
    print_array(x, n);

    printf("Obrtanje:\n");
    reverse_array(x, n);
    print_array(x, n);

    printf("Transformacija:\n");
    transform(x, n);

    printf("Ulevo:\n");
    cycle_left(x, n);
    print_array(x, n);

    printf("Udesno:\n");
    cycle_right(x, n);
    print_array(x, n);
}

```

```

    system("pause");
    return 0;
}

```

Zadatak 5.2 *Funkcije i niske:*

```

#include <stdio.h>
/* Izracunava duzinu datog stringa. */
int duzina(char s[]) {
    int i;
    for(i = 0; s[i] != '\0'; i++);

    return i;
}

/* Obrce dati string. */
void obrni(char s[]) {
    int i, j;

    /*
     Jedan brojac krece od pocetka stringa, drugi od kraja i u svakom
     koraku razmenjuju se elementi sa njihovim indeksima. Pritom se vodi racuna
     da se obrtanje vrši samo dokle se brojac ne sretne, u suprotnom string
     bi se vratio na pocetni.
    */
    for(i = 0, j = duzina(s) - 1; i < j; i++, j--) {
        char c = s[i];
        s[i] = s[j];
        s[j] = c;
    }
}

/* Prevedi neoznaceni ceo broj u binarni zapis. */
void prevedi_u_binarni(unsigned n, char binarni_zapis[]) {
    int i = 0;

    do {
        // U rezultujuci string se upisuje ostatak pri deljenju sa 2.
        binarni_zapis[i++] = (n % 2) + '0';
        // Dati broj se deli sa dva, dokle god se ne dodje do 0.
        n >>= 1; //Operatori nad bitovima ce biti detaljno objasnjeni u nastavku
    } while( n != 0);

    // Na kraju se rezultujuci string terminira 0 i obrce.

```

```

        binarni_zapis[i] = '\0';
        obrni(binarni_zapis);
    }

    /*
    Rekurzivna varijanta prevodjenja u binarni zapis.
    */
    void prevedi_u_binarni_rek(unsigned n) {
        if(n) {
            prevedi_u_binarni_rek(n / 2);
            printf("%u", n % 2);
        }
    }

    int main() {
        unsigned n;

        while(1) {
            printf("Unesite prirodan broj, 0 za kraj unosa: ");
            scanf("%u", &n);

            if( n == 0) {
                break;
            }

            printf("rekurzivno: (");
            prevedi_u_binarni_rek(n);
            printf(")\n");

            char binarni_zapis[32];
            prevedi_u_binarni(n, binarni_zapis);
            printf("nerekurzivno: (%s)\n", binarni_zapis);
        }

        getch();
        return 0;
    }

```

5.2 Operacije nad bitovima - PROGRAMIRANJE NISKOG NIVOAA

Zadatak 5.3 Naglasak u dosadasnjem radu su bile instrukcije i osobine jezika C nezavisne od masine. Manipulacija nad bitovima i ostale operacije niskog nivoa su korisne za sistemsko programiranje (pisanje kompajlera i OS), enkripciju, grafiku i pisanje programa gde je iskoristivost memorije veoma vazna. Ovde

su opisani samo osnovni operatori nad bitovima.

5.2.1 Operatori siftovanja

Simbol	Znacenje
<<	siftovanje u levo
>>	siftovanje u desno

Operandi mogu biti bilo kog celebrojnog tipa ukljucujuci i *char*. Rezultat izraza $i \ll j$ je pomeranje bitova u i za j mesta ulevo. Za svaki bit pomeren ulevo, desno se dodaje bit 0. Rezultat izraza $i \gg j$ je pomeranje bitova u i za j mesta udesno. Ako je i neoznacnog tipa ili je nenegativan, nule se dodaju sa leve strane.

```
unsigned short i, j;
i = 13;      /* i je 13 binarno zapisano: 0000000000001101 */
j = i << 2; /* j dobija vrednost 52:      0000000000110100 */
j = i >> 2; /* j dobija vrednost 3:       0000000000000011 */
```

Ovi operatori ne modifikuju svoje operande. Upotrebom zdruzenih operatora $\ll=$ i $\gg=$ je moguće modifikovati operande:

```
i = 13;      /* i je 13 binarno zapisano: 0000000000001101 */
i <<= 2;     /* i je sada 52:           0000000000110100 */
i >>= 2;     /* i je sada 13:          0000000000001101 */
```

Operatori siftovanja su nizeg prioriteta u odnosu na aritmeticke operatore: $i \ll= 2 + 1$ je isto sto i $i \ll= (2 + 1)$

5.2.2 Ostali operatori nad bitovima

Simbol	Znacenje
~	komplement
&	konjukcija nad bitovima
^	ekskluzivna disjunkcija nad bitovima
	disjunkcija nad bitovima

```
unsigned short i, j, k;
i = 21;      /* i je 21 binarno zapisano: 000000000010101 */
j = 56;      /* j je 56 binarno zapisano: 0000000000111000 */
k = ~i;      /* k je sada 65514 :      1111111111000111 */
k = i & j;    /* k je sada 16:         000000000010000 */
k = i ^ j;    /* k je sada 45:         0000000000101101 */
k = i | j;    /* k je sada 61:         0000000000111101 */
```

5.3 Datoteke

Zadatak 5.4 *Funkcije i nizovi:*