Laslo Kraus

REŠENI ZADACI

IZ

PROGRAMSKOG JEZIKA

C++

(C++20)

P r e d g o v o r

Ova zbirka zadataka je pomoćni udžbenik za učenje programiranja na jeziku *C++*. Zadaci prate gradivo autorove knjige *Programski jezik C++ sa rešenim zadacima* ([1]). Podrazumeva se, kao i u toj knjizi, da je čitalac savladao programiranje na jeziku *C*. Zbirka je namenjena za upotrebu u fakultetskoj nastavi, ali može da se koristi i za samostalno produbljivanje znanja iz programiranja.

Rešenja svih zadataka su potpuna u smislu da priloženi programi mogu da se izvršavaju na računaru. Pored samih tekstova programa priloženo je samo malo objašnjenja, prvenstveno u obliku slika i formula. Očekuje se da će izvođač nastave dati dodatna usmena objašnjenja slušaocima. Uz malo više napora zadaci mogu da se shvate i samostalno. Uz svaki program dat je i primer izvršavanja da bi se olakšalo razumevanje rada programa.

Kroz zadatke, pored elemenata samog jezika, prikazani su osnovni principi objektno-orijentisanog programiranja: sakrivanje podataka, ponovno korišćenje koda, nasleđivanje i polimorfizam. Prikazani su i najčešće korišćeni postupci u programiranju: pretraživanje i uređivanje nizova, obrada znakovnih podataka, rad s bitovima, rad s dinamičkim strukturama podataka (kao što su liste i stabla) i rad s datotekama. Posebna pažnja posvećena je i inženjerskim aspektima programiranja: preglednosti, razumljivosti i efikasnosti.

Jezik *C++* vrlo je složen. Nisu svi detalji neophodni svakome, a naročito ne početnicima. Odeljci u gorepomenutoj knjizi koji mogu da se preskoče u prvom čitanju, bilo zbog složenosti, bilo zbog manjeg značaja, obeleženi su sa △. Zadaci koji koriste elemente jezika *C++* iz tih odeljaka, u ovoj zbirci obeleženi su na isti način. Nezavisno od toga, kao i u knjizi, i ovde su uokvireni delovi programa koje treba izostaviti ako je u knjizi preskočen odeljak „Upućivači na *dvrednosti*" (2.2.11.3). Time se ne gubi ništa na funkcionalnosti programa. Jedino će se neke radnje izvršavati manje efikasno.

Izvorni tekstovi svih programa iz ove zbirke mogu da se preuzmu preko Interneta sa adrese `home.etf.rs/~kraus/knjige/`. Svoja zapažanja čitaoci mogu da upute elektronskom poštom na adresu `kraus@etf.rs`.

Beograd, septembar 2020.

Laslo Kraus

S a d r ž a j

Predgovor **3**

Sadržaj **4**

Preporučena literatura **6**

1 Pripremni zadaci **7**

Zadatak 1.1 Izostavljanje elemenata niza na osnovu binarne maske 8
Zadatak 1.2 Rekurzivno izračunavanje skalarnog proizvoda dva vektora 9
Zadatak 1.3 Presek dva skupa u dinamičkoj zoni memorije 10
Zadatak 1.4 Obrtanje redosleda elemenata jednostruko spregnute liste 11

2 Proširenja jezika C **13**

Zadatak 2.1 Ispisivanje pozdrava 14
Zadatak 2.2 Izračunavanje zbira niza brojeva 15
Zadatak 2.3 Uređivanje dinamičkog niza brojeva 16
Zadatak 2.4 Obrada jednostruko spregnute liste brojeva 17
Zadatak 2.5 Uređivanje niza brojeva metodom podele 18
Zadatak 2.6 Izostavljanje suvišnih razmaka među rečima 20
Zadatak 2.7 Uređivanje imena gradova u dinamičkoj matrici 21
Zadatak 2.8 Određivanje polarnih koordinata tačke 23
Zadatak 2.9 Izračunavanje površine trougla 24
Zadatak 2.10 Paket funkcija za obradu redova brojeva neograničenih kapaciteta 25
Zadatak 2.11 Paketi funkcija za obradu tačaka, pravougaonika i nizova pravougaonika u ravni 27

3 Klase **31**

Zadatak 3.1 Tačke u ravni 32
Zadatak 3.2 Uglovi 33
Zadatak 3.3 Redovi brojeva ograničenih kapaciteta 35
Zadatak 3.4 Uređeni skupovi brojeva 38
Zadatak 3.5 Trouglovi u ravni 41
Zadatak 3.6 Kvadri u dinamičkoj zoni memorije 43
Zadatak 3.7 Krugovi u ravni koji ne smeju da se preklapaju 45
Zadatak 3.8 Kalendarski datumi 48
Zadatak 3.9 Liste brojeva 51
Zadatak 3.10 Uređena stabla brojeva △ 57
Zadatak 3.11 Nizovi materijalnih tačaka 63
Zadatak 3.12 Liste datuma 66
Zadatak 3.13 JMBG, osobe i imenici 69

4 Preopterećivanje operatora....73

Zadatak 4.1 Kompleksni brojevi....74
Zadatak 4.2 Vremenski intervali....76
Zadatak 4.3 Nizovi kompleksnih brojeva....78
Zadatak 4.4 Kvadri s automatski generisanim identifikacionim brojevima....80
Zadatak 4.5 Polinomi s realnim koeficijentima....82
Zadatak 4.6 Studenti koji ne smeju da se kopiraju....86
Zadatak 4.7 Redovi brojeva neograničenih kapaciteta....89
Zadatak 4.8 Tekstovi....92
Zadatak 4.9 Tekstovi s uštedom memorije △....95
Zadatak 4.10 Karte i predstave....99
Zadatak 4.11 Zapisi artikala i inventari....102
Zadatak 4.12 Otpornici i redne veze otpornika....106
Zadatak 4.13 Odrednice i rečnici....109
Zadatak 4.14 Tačke, trouglovi, skupovi trouglova u ravni....112

5 Izvedene klase....117

Zadatak 5.1 Valjci i kante....118
Zadatak 5.2 Osobe, đaci i zaposleni....120
Zadatak 5.3 Knjige, tvrdo koričene, polovne i police....123
Zadatak 5.4 Neuređene i uređene liste celih brojeva....127
Zadatak 5.5 Predmeti, sfere i kvadri....130
Zadatak 5.6 Geometrijske figure, krugovi, kvadrati i trouglovi u ravni....133
Zadatak 5.7 Vektori, brzine i pokretni objekti i tačke u prostoru....138
Zadatak 5.8 Tačke, linije, duži, izlomljene linije i poligoni u ravni....141
Zadatak 5.9 Objekti, skupovi objekata, kompleksni brojevi i tekstovi....147
Zadatak 5.10 Geometrijska tela, sfere, valjci i redovi tela....153
Zadatak 5.11 Osobe, studenti i imenici △....158
Zadatak 5.12 Osobe, vozila, teretna vozila i putnička vozila....162
Zadatak 5.13 Izrazi, konstante, promenljive, dodele vrednosti i aritmetičke operacije △....167
Zadatak 5.14 Naredbe, proste naredbe, sekvence, selekcije i ciklusi △....173

6 Izuzeci....179

Zadatak 6.1 Vektori realnih brojeva sa zadatim opsezima indeksa....180
Zadatak 6.2 Racionalni brojevi....183
Zadatak 6.3 Matrice racionalnih brojeva △....186
Zadatak 6.4 Nizovi, funkcije i verižni razlomci....192
Zadatak 6.5 Podaci, skalarni podaci i nizovi....196
Zadatak 6.6 Funkcije i greške; izračunavanje određenog integrala....201
Zadatak 6.7 Predmeti, celi brojevi, zbirke i nizovi predmeta....207
Zadatak 6.8 Vektori, figure, tačke i mnogouglovi u prostoru....212
Zadatak 6.9 Proizvodi, sfere, kvadri, mašine i radnici....217
Zadatak 6.10 Radnici, prodavci, šefovi i firme....223
Zadatak 6.11 Vozila, lokomotive, putnički vagoni i vozovi....228
Zadatak 6.12 Električni potrošači, uređaji, grupe uređaja i izvori....234
Zadatak 6.13 Funkcije za koje mogu da se stvaraju izvodi; delegati, monomi, eksponencijalne funkcije i zbirovi funkcija △....239

7 Generičke stvari....245

Zadatak 7.1 Generička funkcija za fuziju uređenih nizova....246
Zadatak 7.2 Generički stekovi zadatih kapaciteta....249

Zadatak 7.3 Generičke klase za upoređivanje podataka i uređivanje nizova podataka 251
Zadatak 7.4 Generički nizovi, boje, tačke, obojene figure, krugovi, pravougaonici, trouglovi, mnogouglovi i crteži u ravni 253
Zadatak 7.5 Generičke liste, datumi, osobe, ispiti, đaci i škole 262
Zadatak 7.6 Vozila, bicikli, kamioni, generički nizovi, etape, vožnje i trkački automobili 268
Zadatak 7.7 Tereti, sanduci, burad, generički nizovi, generički pametni pokazivači, vozila, lokomotive, vagoni i vozovi △ 274
Zadatak 7.8 Simboli, fontovi, vektori, duži, tekstovi, generički nizovi i crteži △ 282
Zadatak 7.9 Akteri, časovnici, proizvodi i generička skladišta, proizvođači i potrošači △ 290

8 Lambda izrazi △ 297

Zadatak 8.1 Nalaženje minimuma funkcije △ 298
Zadatak 8.2 Grafički prikaz funkcija △ 299

9 Standardna biblioteka △ 303

Zadatak 9.1 Stekovi i redovi tačaka neograničenih kapaciteta △ 304
Zadatak 9.2 Predmeti, tela, sfere, kvadri i sklopovi △ 308

10 Ulaz i izlaz 315

Zadatak 10.1 Obrada sekvencijalne tekstualne datoteke 316
Zadatak 10.2 Obrada rečenica u tekstualnoj sekvencijalnoj datoteci 317
Zadatak 10.3 Obrada sekvencijalne binarne datoteke 318
Zadatak 10.4 Obrada relativne binarne datoteke 320
Zadatak 10.5 Klasa rečnika △ 322
Zadatak 10.6 Klasa relativnih binarnih datoteka i obrada liste u relativnoj binarnoj datoteci △ 326

Preporučena literatura

[1] Laslo Kraus: **Programski jezik C++ sa rešenim zadacima**, *C++17*, jedanaesto izdanje, Akademska misao, Beograd, *2019*.

[2] **Working Draft, Standard for Programming Language C++**, ISO/IEC, WG21 Number N4849, http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/, *2020.*

[3] **C++ reference**, https://en.cppreference.com/w/cpp, *2020.*

[4] Bjarne Stroustrup: **The C++ Programming Language**, *Third Edition*, Addison-Wesley Publishing Company, Reading, Massachusetts, *1997*.

[5] Laslo Kraus: **Programski jezik C sa rešenim zadacima**, *C18*, deseto izdanje, Akademska misao, Beograd, *2020*.

[6] Laslo Kraus: **Rešeni zadaci iz programskog jezika C**, *C18*, peto izdanje, Akademska misao, Beograd, *2020*.

1 Pripremni zadaci

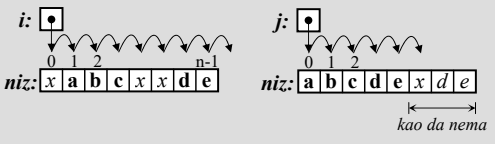
Zadatak 1.1 *Izostavljanje elemenata niza na osnovu binarne maske*

Napisati na jeziku *C* funkciju za izostavljanje svih elemenata numeričkog niza `niz` dužine `n` elemenata za koje na odgovarajućim mestima niza bitova `maska` stoji 0. Niz bitova se smešta u potreban broj bajtova od kojih svaki (sem možda poslednjeg) sadrži po 8 bitova.

Napisati na jeziku *C* program za ispitivanje prethodne funkcije.

Rešenje:

Izostavljanje određenih elemenata:
- Elementi se ispituju redom (indeks *i*). Ako *i*-ti element treba da ostane, premešta se na *j*-to mesto.
- Nova dužina niza je završna vrednost indeksa *j*.



a) pre obrade *b) posle obrade*

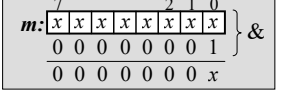
```
// reduk.c - Izostavljanje elemenata niza na osnovu maske.

void redukcija(int niz[], const char maska[], int *n) {
  int i, j, k; char m;
  for (i=j=k=0; i<*n; i++) {
    if (i % 8 == 0) m = maska[k++];
    if (m & 1) niz[j++] = niz[i];
    m >>= 1;
  }
  *n = j;
}

#include <stdio.h>

int main() {
  enum {N = 120};
  while (1) {
    printf("n? "); int n; scanf("%d", &n);
  if (n<=0 || n>N) break;
    printf("niz? "); int niz[120];
    for (int i=0; i<n; scanf("%d", &niz[i++]));
    printf("maska? "); char maska[(N+7)/8];
    for (int i=0; i<(n+7)/8; i++) {
      int m; scanf("%x", &m); maska[i] = m;
    }
    redukcija(niz, maska, &n);
    printf("niz=");
    for (int i=0; i<n;
         printf(" %d", niz[i++]));
    putchar('\n');
  }
}
```

Izdvajanje desnog bita:




```
% reduk
n? 11
niz? 1 2 3 4 5 6 7 8 9 0 1
maska? b5 4
niz= 1 3 5 6 8 1
n? 8
niz? 1 2 3 4 5 6 7 8
maska? ff
niz= 1 2 3 4 5 6 7 8
n? 5
niz? 1 2 3 4 5
maska? 0
niz=
n? 0
```

Zadatak 1.2 *Rekurzivno izračunavanje skalarnog proizvoda dva vektora*

Napisati na jeziku *C* rekurzivnu funkciju za izračunavanje skalarnog proizvoda dva vektora.

Napisati na jeziku *C* program koji pročita dva vektora sa standardnog ulaza, izračuna njihov skalarni proizvod, ispiše dobijeni rezultat na standardnom izlazu i ponavlja prethodne korake sve dok ne dobije signal za završetak rada programa. Program treba da je pogodan za skretanje standardnog ulaza na datoteku.

Rešenje:

```
// skalpro.c - Skalarni proizvod dva vektora.

double skal_pro(const double *a, const double *b, int n) {
  return n ? (*a) * (*b) + skal_pro(a+1, b+1, n-1) : 0;
}

#include <stdio.h>

int main() {
  int n;
  while (scanf("%i", &n) != EOF) {
    double a[100];
    for (int i=0; i<n; scanf("%lf", &a[i++]));
    double b[100];
    for (int i=0; i<n; scanf("%lf", &b[i++]));
    printf("%g\n", skal_pro(a,b,n));
  }
}
```

sklapro.pod

```
5
1 2 3 4 5
5 4 3 2 1
4
-1 1 -2 2
4 5 6 7
```

```
% skalpro <skalpro.pod >skalpro.rez
```

skalpro.rez

```
35
3
```

Zadatak 1.3 *Presek dva skupa u dinamičkoj zoni memorije*

Skup realnih brojeva predstavlja se pomoću strukture od dva člana koji su broj elemenata skupa i pokazivač na niz u dinamičkoj zoni memorije koji sadrži same elemente skupa. Napisati na jeziku *C* funkciju za nalaženje preseka dva takva skupa.

Napisati na jeziku *C* program koji pročita dva skupa realnih brojeva, pronalazi njihov presek, ispisuje dobijeni rezultat i ponavlja prethodne korake sve dok za broj elemenata skupa ne pročita negativnu vrednost.

Rešenje:

```
// presek.c - Presek dva skupa.

#include <stdio.h>
#include <stdlib.h>

typedef struct { int vel; double *niz; } Skup;

Skup presek(Skup s1, Skup s2) {
  int vel = (s1.vel<s2.vel ? s1.vel : s2.vel);
  double *niz = malloc(vel*sizeof(double));
  int k = 0;
  for (int i=0; i<s1.vel; i++) {
    int j = 0; while (j<s2.vel && s2.niz[j]!=s1.niz[i]) j++;
    if (j < s2.vel) niz[k++] = s2.niz[j];
  }
  Skup s = {k, realloc(niz, k*sizeof(double))};
  return s;
}

int main() {
  while (1) {
    Skup s1; scanf("%d", &s1.vel);
  if (s1.vel < 0) break;
    s1.niz = malloc(s1.vel*sizeof(double));
    for (int i=0; i<s1.vel; scanf("%lf",&s1.niz[i++]));
    Skup s2; scanf("%d", &s2.vel);
  if (s2.vel < 0) break;
    s2.niz = malloc(s2.vel*sizeof(double));
    for (int i=0; i<s2.vel; scanf("%lf",&s2.niz[i++]));
    Skup s = presek(s1, s2);
    for (int i=0; i<s.vel; printf("%lf ", s.niz[i++])); putchar('\n');
    free(s1.niz); free(s2.niz); free(s.niz);
  }
}
```

presek.pod

```
4   1 2 3 4
5   3 4 5 6 7
6   9 3 5 1 2 8
4   6 2 8 5
-1
```

```
% presek <presek.pod
3.000000 4.000000
5.000000 2.000000 8.000000
```