PROGRAMIRANJE

ZBIRKA ZADATAKA

Ishod učenja 5

2023-2024.

Zadatak 1

Napišite strukturu KucniLjubimac sa članovima ime i vrsta. Napišite program koji korisnika pita koliko kućnih ljubimaca želi učitati. Na osnovu upisane vrijednosti pripremite dinamičku memoriju te u nju učitajte objekte od korisnika. Na kraju, ispišite sve učitane kućne ljubimce.

Moguće rješenje:

```
#include <iostream>
#include <string>
using namespace std;

struct KucniLjubimac {
	string ime;
	string vrsta;
};

int main() {
	int koliko;
	cout << "Koliko kucnih ljubimaca zelite upisati: ";
	cin >> koliko;
	cin.ignore();

	KucniLjubimac* ljubimci = new KucniLjubimac[koliko];

	for (int i = 0; i < koliko; i++) {
		cout << "Unos " << (i + 1) << ". ljubimca" << endl;

		cout << "Ime: ";
		getline(cin, ljubimci[i].ime);
		cout << "Vrsta: ";
		getline(cin, ljubimci[i].vrsta);
	}

	cout << "Popis kucnih ljubimaca" << endl;
	cout << "----------------------" << endl;
	for (int i = 0; i < koliko; i++) {
		cout << ljubimci[i].ime << " (" << ljubimci[i].vrsta << ")" << endl;
	}

	delete[] ljubimci;

	return 0;
}
```

Zadatak 2

Napišite funkciju koja prima polje s dnevnom količinom padalina te izračunava i vraća pozivatelju prosječnu i najveću količinu padalina (način vraćanja dvije vrijednosti iz funkcije odaberite sami, bitno je da se obje vrijednosti dobiju jednim pozivom funkcije). U polju padalina negativne vrijednosti predstavljaju pogrešku mjerenja pa ih samo ignorirajte. Ako su sve vrijednosti u polju negativne, vratite -1 i za prosjek i za najveću količinu. U main()-u pozovite funkciju s poljem veličine 5 i ispišite obje vraćene vrijednosti. Primjerice, ako se funkciji pošalje polje 0, 0.6, 1.1, -1, 0, funkcija treba vratiti prosjek od 0.425 i najveću vrijednost od 1.1.

Moguće rješenje:

```
#include <iostream>
using namespace std;

void izracunaj(double padaline[], int n, double& prosjecna, double& najveca) {
	double max_vrijednost = padaline[0];
	double zbroj = 0.0;
	int kolicina_nenegativnih = 0;

	for (int i = 0; i < n; i++) {
		if (padaline[i] >= 0.0) {
			zbroj += padaline[i];

			if (padaline[i] > max_vrijednost) {
				max_vrijednost = padaline[i];
			}

			kolicina_nenegativnih++;
		}
	}

	if (kolicina_nenegativnih > 0) {
		prosjecna = zbroj / kolicina_nenegativnih;
		najveca = max_vrijednost;
	}
	else {
		prosjecna = -1.0;
		najveca = -1.0;
	}
}

int main() {
	const int n = 5;
	double p[n] = { 0, 0.6, 1.1, -1, 0 };
	double prosjecna;
	double najveca;

	izracunaj(p, n, prosjecna, najveca);

	cout << "Prosjecna: " << prosjecna << endl;
	cout << "Najveca: " << najveca << endl;
	return 0;
}
```

Zadatak 3

Napišite funkciju koja prima polje cijelih brojeva (veličina po želji) i vektor te brojeve iz polja kopira u vektor, ali tako da se neki broj smije najviše dva puta pojaviti u vektoru. U main()-u pripremite polje i vektor, pozovite funkciju te nakon toga ispišite sadržaj vektora. Primjerice, ako imamo polje: 1, 2, 1, 1, 1, 1, 1, 3, u vektor treba staviti: 1, 2, 1, 3. Slobodno kreirajte dodatne funkcije prema želji.

Moguće rješenje:

```
#include <iostream>
#include <vector>
using namespace std;

int broj_pojavljivanja(vector<int> v, int broj) {
	int n = 0;
	for (int i = 0; i < v.size(); i++) {
		if (v[i] == broj) {
			n++;
		}
	}
	return n;
}

void kopiraj(int brojevi[], int n, vector<int>& jedinstveni) {
	for (int i = 0; i < n; i++) {
		if (broj_pojavljivanja(jedinstveni, brojevi[i]) < 2) {
			jedinstveni.push_back(brojevi[i]);
		}
	}
}

int main() {
	const int n = 8;
	int p[n] = { 1, 2, 1, 1, 1, 1, 1, 3 };
	vector<int> v;

	kopiraj(p, n, v);

	for (int i = 0; i < v.size(); i++) {
		cout << v[i] << " ";
	}

	return 0;
}
```

Zadatak 4

Napišite program koji prvo pita korisnika koliko godina želi generirati. Neka program zatim generira toliko slučajnih godina u rasponu od 1945. do 2023. i spremi ih u dinamičku memoriju. Neka program zatim učita godinu od korisnika, te ispiše sve godine iz dinamičke memorije koje su veće ili jednake učitanoj.

Moguće rješenje:

```
#include <iostream>
#include <ctime>
using namespace std;

int main() {
	srand(time(nullptr));

	int n;
	cout << "Koliko godina zelite generirati: ";
	cin >> n;

	int* godine = new int[n];

	for (int i = 0; i < n; i++) {
		godine[i] = 1945 + rand() % (2023 - 1945 + 1);
	}

	int min_godina;
	cout << "Upisite godinu: ";
	cin >> min_godina;

	for (int i = 0; i < n; i++) {
		if (godine[i] >= min_godina) {
			cout << godine[i] << " ";
		}
	}

	delete[] godine;
	return 0;
}
```

Zadatak 5

Napišite jednu funkciju koja prima dva cjelobrojna parametra i zamijeni im vrijednosti. Napišite i drugu funkciju koja prima polje cijelih brojeva i ispisuje ih u jednom retku odvojene razmacima. U main-u napravite polje od pet cijelih brojeva. Nakon toga, pozivajte prvu funkciju za svaka dva susjedna broja (nulti i prvi, pa prvi i drugi, itd.) Nakon svakog poziva prve funkcije, pozovite i drugu funkciju kako biste ispisali sadržaj polja. Primjer rada programa:

```
Pocetno stanje: 122 55 34 72 18
55 122 34 72 18
55 34 122 72 18
55 34 72 122 18
55 34 72 18 122
```

Moguće rješenje:

```cpp
#include <iostream>
#include <string>
using namespace std;

void zamijeni(int& a, int& b) {
	int temp = a;
	a = b;
	b = temp;
}

void ispisi(int brojevi[], int n) {
	for (int i = 0; i < n; i++) {
		cout << brojevi[i] << " ";
	}
	cout << endl;
}

int main() {
	const int n = 5;
	int brojevi[n] = { 122, 55, 34, 72, 18 };

	cout << "Pocetno stanje: ";
	ispisi(brojevi, n);

	for (int i = 0; i < n - 1; i++) {
		zamijeni(brojevi[i], brojevi[i + 1]);
		ispisi(brojevi, n);
	}

	return 0;
}
```

Zadatak 6

Napišite program pita korisnika koliko imena želi učitati. Nakon toga, učitajte željeni broj imena u dinamičku memoriju. Zatim, prekopirajte sva imena koja završavaju na slovo 'a' u novi dio dinamičke memorije (pazite, može se dogoditi da nema niti jedno takvo ime). Na kraju, ispišite sva imena koja završavaju sa slovom 'a' na ekran.

Moguće rješenje:

```
#include <iostream>
#include <string>
using namespace std;

int main() {
	// pitamo korisnika koliko imena zeli ucitati
	int koliko;
	cout << "Koliko imena zelite ucitati: ";
	cin >> koliko;
	cin.ignore();

	// ucitamo sva imena u dinamicko polje
	string* sva_imena = new string[koliko];
	for (int i = 0; i < koliko; i++) {
		cout << "Upisite ime: ";
		getline(cin, sva_imena[i]);
	}

	// prebrojimo koliko imena zavrsava sa slovom 'a'
	int kolicina = 0;
	for (int i = 0; i < koliko; i++) {
		if (sva_imena[i][sva_imena[i].length() - 1] == 'a') {
			kolicina++;
		}
	}

	// ako nema niti jednog imena, kraj
	if (kolicina == 0) {
		cout << "Nema niti jednog imena koje zavrsava na slovo 'a'" << endl;
		return 0;
	}

	// prekopiramo takva imena u novo dinamicko polje
	string* odabrana_imena = new string[kolicina];
	int trenutni_index = 0;

	for (int i = 0; i < koliko; i++) {
		if (sva_imena[i][sva_imena[i].length() - 1] == 'a') {
			odabrana_imena[trenutni_index] = sva_imena[i];
			trenutni_index++;
		}
	}

	// otpustimo originalno dinamicko polje jer nam vise ne treba
	delete[] sva_imena;

	// ispisemo imena koja zavrsavaju na slovo 'a'
```

```
	cout << "Imena koja zavrsavaju na slovo 'a':" << endl;
	for (int i = 0; i < kolicina; i++) {
		cout << odabrana_imena[i] << endl;
	}

	// otpustimo i drugo polje
	delete[] odabrana_imena;

	return 0;
}
```

Zadatak 7

Napišite funkciju koja prima vektor i njega učitava tri cijene. Napišite i funkciju koja prima vektor i izračunava i vraća prosječnu cijenu svih cijena iz vektora. U main-u pripremite prazni vektor s cijenama. Omogućite korisniku opcije učitavanja nove tri cijene u vektor te ispis prosječne cijene svih cijena trenutno u vektoru. Ponavljajte dok to korisnik želi. Primjer rada programa:

```
A = ucitati tri nove cijene u vektor
B = ispisati prosjecnu cijenu
X = izlaz
> A
Cijena: 10
Cijena: 10
Cijena: 10
> B
10
> A
Cijena: 11
Cijena: 12.5
Cijena: 14
> B
11.25
> X
```

Moguće rješenje:

```
#include <iostream>
#include <vector>
using namespace std;

void ucitaj(vector<double>& cijene) {
	double cijena;
	for (int i = 0; i < 3; i++) {
		cout << "Cijena: ";
		cin >> cijena;
		cijene.push_back(cijena);
	}
}

double prosjek(vector<double>& cijene) {
	double suma = 0.0;
	for (int i = 0; i < cijene.size(); i++) {
		suma += cijene[i];
	}
	return suma / cijene.size();
}

int main() {
	vector<double> cijene;
	char opcija;

	cout << "A = ucitati tri nove cijene u vektor" << endl;
	cout << "B = ispisati prosjecnu cijenu" << endl;
	cout << "X = izlaz" << endl;

	do {
```

```
            cout << "> ";
            cin >> opcija;

            switch (opcija) {
            case 'A':
                  ucitaj(cijene);
                  break;
            case 'B':
                  cout << prosjek(cijene) << endl;
                  break;
            case 'X':
                  break;
            default:
                  cout << "Nepoznata opcija" << endl;
                  break;
            }
      } while (opcija != 'X');

      return 0;
}
```

Zadatak 8

Definirajte strukturu Pravokutnik. Napišite funkciju koja u dinamičkoj memoriji kreira jedan pravokutnik sa slučajnom širinom i visinom između 1 i 10 te njegovu adresu vraća pozivatelju. Pozovite funkciju iz main-a te nakon toga ispišite površinu pravokutnika.

Moguće rješenje:

```
#include <iostream>
#include <ctime>
using namespace std;

struct Pravokutnik {
	int a;
	int b;
};

Pravokutnik* kreiraj() {
	Pravokutnik* p = new Pravokutnik;
	p->a = 1 + rand() % (10 - 1 + 1);
	p->b = 1 + rand() % (10 - 1 + 1);

	return p;
}

int main() {
	srand(time(nullptr));

	Pravokutnik* pravokutnik = kreiraj();
	int povrsina = pravokutnik->a * pravokutnik->b;

	cout << "Povrsina pravokutnika je: " << povrsina << endl;

	delete pravokutnik;
	return 0;
}
```

Zadatak 9

Definirajte strukturu Pravokutnik. Napišite funkciju koja u dinamičkoj memoriji kreira jedan pravokutnik sa slučajnom širinom i visinom između 1 i 10 te njegovu adresu vraća pozivatelju. Iskoristite funkciju iz main-a tako da napravite vektor od 100 dinamičkih pravokutnika. Na kraju ispišite površine svih pravokutnika.

Moguće rješenje:

```
#include <iostream>
#include <ctime>
#include <vector>
using namespace std;

struct Pravokutnik {
	int a;
	int b;
};

Pravokutnik* kreiraj() {
	Pravokutnik* p = new Pravokutnik;
	p->a = 1 + rand() % (10 - 1 + 1);
	p->b = 1 + rand() % (10 - 1 + 1);

	return p;
}

int main() {
	srand(time(nullptr));

	vector<Pravokutnik*> pravokutnici;

	for (int i = 0; i < 100; i++) {
		Pravokutnik* p = kreiraj();
		pravokutnici.push_back(p);
	}

	for (int i = 0; i < pravokutnici.size(); i++) {
		int povrsina = pravokutnici[i]->a * pravokutnici[i]->b;
		cout << "Povrsina pravokutnika je: " << povrsina << endl;
	}

	for (int i = 0; i < pravokutnici.size(); i++) {
		delete pravokutnici[i];
	}

	return 0;
}
```

Zadatak 10

Napišite program koji korisnika pita koliko brojeva želi učitati. Na osnovu upisane vrijednosti pripremite dinamičku memoriju te u nju učitajte brojeve od korisnika. Na kraju pronađite i ispišite najmanji uneseni broj.

Moguće rješenje:

```
#include <iostream>
using namespace std;

int main() {
	int koliko;
	cout << "Koliko brojeva zelite ucitati: ";
	cin >> koliko;

	int* brojevi = new int[koliko];

	for (int i = 0; i < koliko; i++) {
		cout << "Broj: ";
		cin >> brojevi[i];
	}

	int najmanji = brojevi[0];

	for (int i = 0; i < koliko; i++) {
		if (brojevi[i] < najmanji) {
			najmanji = brojevi[i];
		}
	}

	cout << "Najmanji broj je: " << najmanji << endl;

	delete[] brojevi;
	return 0;
}
```

Zadatak 11

Napišite funkciju `presjek` koja prima dva polja cijelih brojeva i vektor te one brojeve iz prvog polja koji se pojavljuju i u drugom polju dodaje u vektor. U main-u pripremite oba polja i vektor, pozovite funkciju te nakon toga ispišite sadržaj vektora. Primjerice, ako je prvo polje 1, 2, 3, 4, 5 i drugo polje 4, 4, 5, 6, 7, u vektor treba staviti 4, 5.

Moguće rješenje:

```
#include <iostream>
#include <vector>
using namespace std;

bool dio_polja(int broj, int p[], int n) {
	for (int i = 0; i < n; i++) {
		if (p[i] == broj) {
			return true;
		}
	}

	return false;
}

void presjek(int polje1[], int n1, int polje2[], int n2, vector<int>& v) {
	for (int i = 0; i < n1; i++) {
		if (dio_polja(polje1[i], polje2, n2)) {
			v.push_back(polje1[i]);
		}
	}
}

int main() {
	int p1[5] = { 1, 2, 3, 4, 5 };
	int p2[5] = { 4, 4, 5, 6, 7 };
	vector<int> v;

	presjek(p1, 5, p2, 5, v);

	for (int i = 0; i < v.size(); i++) {
		cout << v[i] << " ";
	}

	return 0;
}
```

Zadatak 12

Napišite program koji od korisnika učitava 5 imena. Neka program sva imena koja imaju 4 ili manje znakova prepiše u dinamičko polje. Na kraju ispišite sadržaj dinamičkog polja. Smijete pretpostaviti da će korisnik unijeti barem jedno ime sa 4 ili manje slova.

Moguće rješenje:

```
#include <iostream>
#include <string>
using namespace std;

int main() {
	const int n = 5;
	string imena[n];
	for (int i = 0; i < n; i++) {
		cout << "Ime: ";
		getline(cin, imena[i]);
	}

	int broj_imena_s_manje_od_5_slova = 0;
	for (int i = 0; i < n; i++) {
		if (imena[i].length() < 5) {
			broj_imena_s_manje_od_5_slova++;
		}
	}

	string* kratka = new string[broj_imena_s_manje_od_5_slova];
	int indeks_din_polja = 0;

	for (int i = 0; i < n; i++) {
		if (imena[i].length() < 5) {
			kratka[indeks_din_polja] = imena[i];
			indeks_din_polja++;
		}
	}

	cout << "Kratka imena" << endl;
	for (int i = 0; i < broj_imena_s_manje_od_5_slova; i++) {
		cout << kratka[i] << " ";
	}

	delete[] kratka;
	return 0;
}
```

Zadatak 13

Napišite funkciju koja prima dva polja znakova: u prvo je upisano 11 znakova, a drugo ima istu veličinu, ali je prazno. Funkcija treba iz prvog polja prepisati sve znakove u drugo, ali u obrnutom redoslijedu. U main-u pripremite polja, pozovite funkciju te ispišite sadržaj drugog polja.

Moguće rješenje:

```
#include <iostream>
using namespace std;

void obrni(char polje1[], char polje2[], int velicina_svakog_polja) {
	for (int i = 0; i < velicina_svakog_polja; i++) {
		polje2[velicina_svakog_polja - 1 - i] = polje1[i];
	}
}

int main() {
	const int n = 11;
	char prvo[n] = { '1', '0', '0', '0', '0', 'Z', 'a', 'g', 'r', 'e', 'b' };
	char drugo[n];

	obrni(prvo, drugo, n);

	for (int i = 0; i < n; i++) {
		cout << drugo[i];
	}
	cout << endl;

	return 0;
}
```

Zadatak 14

Napišite program u kojemu vektor cijelih brojeva glumi bazu podataka. Omogućite sljedeće operacije, svaku u svojoj funkciji i ponavljajte dok to korisnik želi:

- Unos novog broja.
- Ažuriranje svih pojavljivanja nekog broja na novu vrijednost.
- Ispis svih pojavljivanja nekog broja.
- Ispis kompletne baze podataka.

Primjer rada programa:

```
1 = unos novog broja
2 = azuriranje svih pojavljivanja nekog broja na novu vrijednost
3 = ispis svih pojavljivanja nekog broja
4 = ispis kompletne baze podataka
5 = kraj
> 1
Upisite novi broj: 1825
Broj uspjesno dodan u bazu podataka
> 1
Upisite novi broj: 1954
Broj uspjesno dodan u bazu podataka
> 1
Upisite novi broj: 1825
Broj uspjesno dodan u bazu podataka
> 4
1825 1954 1825
> 2
Upisite broj koji zelite azurirati: 1825
Upisite novu vrijednost: 1827
Brojevi uspjesno azurirani
> 3
Upisite broj koji zelite ispisati: 1827
1827 1827
> 4
1827 1954 1827
> 5
```

Moguće rješenje:

```
#include <iostream>
#include <vector>
using namespace std;

void unesi_novi(vector<int>& baza) {
	int broj;
	cout << "Upisite novi broj: ";
	cin >> broj;
	baza.push_back(broj);
	cout << "Broj uspjesno dodan u bazu podataka" << endl;
}

void azuriraj(vector<int>& baza) {
	int broj_stari;
	cout << "Upisite broj koji zelite azurirati: ";
```

```
	cin >> broj_stari;

	int broj_novi;
	cout << "Upisite novu vrijednost: ";
	cin >> broj_novi;

	for (int i = 0; i < baza.size(); i++) {
		if (baza[i] == broj_stari) {
			baza[i] = broj_novi;
		}
	}
	cout << "Brojevi uspjesno azurirani" << endl;
}

void ispisi_pojavljivanje(vector<int>& baza) {
	int broj;
	cout << "Upisite broj koji zelite ispisati: ";
	cin >> broj;

	for (int i = 0; i < baza.size(); i++) {
		if (baza[i] == broj) {
			cout << baza[i] << " ";
		}
	}
	cout << endl;
}

void ispisi_sve(vector<int>& baza) {
	for (int i = 0; i < baza.size(); i++) {
		cout << baza[i] << " ";
	}
	cout << endl;
}

int main() {
	vector<int> baza;
	char opcija;

	cout << "1 = unos novog broja" << endl;
	cout << "2 = azuriranje svih pojavljivanja nekog broja na novu vrijednost" <<
endl;
	cout << "3 = ispis svih pojavljivanja nekog broja" << endl;
	cout << "4 = ispis kompletne baze podataka" << endl;
	cout << "5 = kraj" << endl;

	do {
		cout << "> ";
		cin >> opcija;

		switch (opcija) {
		case '1': unesi_novi(baza); break;
		case '2': azuriraj(baza); break;
		case '3': ispisi_pojavljivanje(baza); break;
		case '4': ispisi_sve(baza); break;
		case '5': break;
		default: cout << "neispravna opcija" << endl; break;
		}
	} while (opcija != '5');
```

```
        return 0;
}
```

Zadatak 15

Genom je kompletan skup uputa koje živi organizam treba za funkcioniranje i sastoji se od stringa sastavljenog od slova A, T, C i G. Napišite funkciju koja pozivatelju vraća string sastavljen od 40 slučajnih vrijednosti A, T, C i G (za dodavanje novog znaka postojećem stringu koristite operator +). Kreirajte u dinamičkoj memoriji onoliko genoma koliko to korisnik želi i zatim ih ispišite. Primjer rada programa:

```
Koliko genoma: 7
Genom 1: TGCATACCCATTTGTGGCGAGACTACTACGGCGCTAGGGC
Genom 2: GGCTTATGTAGACTTGGTTCAGCCACACAACTCTCCCTTA
Genom 3: TGACAACAGAAGCGTCCCTTTGGCACTCTCTTAACTTTTA
Genom 4: CCTAGGGCGACCTACCGGGTTGTGTTTCTACCCCTAAGCG
Genom 5: CGATAGGTCCGACAATAGAATCTGCCAAGGATAGGTCTTC
Genom 6: TAGGATGACGAAATTCGTTGGGCGAACCAACTTCTTACCA
Genom 7: CGCTCGGAAGCCAGACCCGTTCAATCCGTGTAATGAACGT
```

Moguće rješenje:

```
#include <iostream>
#include <string>
using namespace std;

string generiraj_genom() {
	string rezultat = "";
	char novi_znak;
	int slucajni_broj;

	for (int i = 0; i < 40; i++) {
		switch (rand() % 4) {
		case 0: novi_znak = 'A'; break;
		case 1: novi_znak = 'T'; break;
		case 2: novi_znak = 'C'; break;
		case 3: novi_znak = 'G'; break;
		}
		rezultat += novi_znak;
	}

	return rezultat;
}

int main() {
	int n;
	cout << "Koliko genoma: ";
	cin >> n;
	string* genomi = new string[n];

	for (int i = 0; i < n; i++) {
		genomi[i] = generiraj_genom();
	}

	for (int i = 0; i < n; i++) {
		cout << "Genom " << (i + 1) << ": " << genomi[i] << endl;
	}

	delete[] genomi;
	return 0;
}
```

Zadatak 16

U main-u kreirajte tri vektora: neka jedan sadrži nekoliko imenica, drugi nekoliko pridjeva, a treći nekoliko glagola. Napišite funkciju koja prima tri vektora te vraća rečenicu sastavljenu od: slučajni pridjev + slučajna imenica + slučajni glagol + slučajni pridjev + slučajna imenica. Pozovite funkciju pet puta i ispišite dobivene rečenice. Primjer rada programa:

```
iskren odmor letjeti plav odmor.
lijep sunce je iskren pas.
plav pas letjeti odan sunce.
iskren odmor letjeti plav pas.
iskren sunce pjevati lijep pas.
```

Moguće rješenje:

```
#include <iostream>
#include <vector>
#include <string>
#include <ctime>
using namespace std;

string generiraj_recenicu(vector<string>& imenice, vector<string>& pridjevi,
vector<string>& glagoli) {
	string recenica = "";

	recenica += pridjevi[rand() % pridjevi.size()];
	recenica += " ";
	recenica += imenice[rand() % imenice.size()];
	recenica += " ";
	recenica += glagoli[rand() % glagoli.size()];
	recenica += " ";
	recenica += pridjevi[rand() % pridjevi.size()];
	recenica += " ";
	recenica += imenice[rand() % imenice.size()];
	recenica += ".";

	return recenica;
}

int main() {
	srand(time(nullptr));
	vector<string> imenice = { "odmor", "nebo", "sunce", "oblak", "pas" };
	vector<string> pridjevi = { "plav", "lijep", "bijel", "iskren", "odan" };
	vector<string> glagoli = { "je", "letjeti", "pjevati" };

	for (int i = 0; i < 5; i++) {
		cout << generiraj_recenicu(imenice, pridjevi, glagoli) << endl;
	}

	return 0;
}
```

Zadatak 17

Napišite program koji radi sa jednodimenzionalnim silama, odnosno, sa silama koje mogu djelovati ili u lijevom smjeru ili u desnom smjeru. Definirajte strukturu Sila koja sadrži članove smjer (lijevo, desno ili nigdje) i iznos. Napišite funkciju koja prima polje sila i pozivatelju vraća (kojim god mehanizmom želite) rezultirajuću silu (dobije se zbrajanjem svih sila iz polja. Primjerice, ako u polju imamo silu A koja djeluje sa 10 N u lijevo, silu B koja djeluje sa 5 N u desno i silu C koja djeluje sa 7 N u desno, onda rezultirajuća sila iznos 2 N u desno). U main-u napravite dinamičko polje s tri sile, pošaljite ga u funkciju i ispišite vraćenu rezultirajuću silu.

Moguće rješenje:

```
#include <iostream>
#include <vector>
#include <string>
#include <ctime>
using namespace std;

struct Sila {
	char smjer; /* L, 0 ili D */
	int iznos;
};

void izracunaj_rezultirajucu_silu(Sila sile[], int n, Sila& rezultirajuca) {
	for (int i = 0; i < n; i++) {
		if (sile[i].smjer == 'L') {
			rezultirajuca.iznos -= sile[i].iznos;
		}
		else if (sile[i].smjer == 'D') {
			rezultirajuca.iznos += sile[i].iznos;
		}
	}

	if (rezultirajuca.iznos < 0) {
		rezultirajuca.iznos *= -1;
		rezultirajuca.smjer = 'L';
	}
	else if (rezultirajuca.iznos > 0) {
		rezultirajuca.smjer = 'D';
	}
	else {
		rezultirajuca.smjer = '0';
	}
}

int main() {
	int n = 3;
	Sila* sile = new Sila[n];

	sile[0].iznos = 10;
	sile[0].smjer = 'L';

	sile[1].iznos = 5;
	sile[1].smjer = 'D';
```

```
	sile[2].iznos = 7;
	sile[2].smjer = 'D';

	Sila rezultirajuca;
	rezultirajuca.iznos = 0;
	rezultirajuca.smjer = '0';

	izracunaj_rezultirajucu_silu(sile, n, rezultirajuca);

	cout << "Rezultirajuca sila djeluje u " << rezultirajuca.smjer << " iznosom
od " << rezultirajuca.iznos << " N" << endl;

	delete[] sile;
	return 0;
}
```

Zadatak 18

Napišite funkciju koji prima polje cijelih brojeva i dva vektora. Neka funkcija sve parne brojeve iz polja kopira u prvi vektor, a sve neparne u drugi vektor. Negativne brojeve i nulu ignorirajte, tj. nemojte prekopirati niti u jedan vektor. U main-u pripremite polje i vektore, pozovite funkciju, a zatim ispišite prvo sve parne, a zatim sve neparne brojeve.

Moguće rješenje:

```
#include <iostream>
#include <vector>
#include <string>
#include <ctime>
using namespace std;

void razvrstaj(int brojevi[], int n, vector<int>& parni, vector<int>& neparni) {
	for (int i = 0; i < n; i++) {
		if (brojevi[i] > 0) {
			if (brojevi[i] % 2 == 0) {
				parni.push_back(brojevi[i]);
			}
			else {
				neparni.push_back(brojevi[i]);
			}
		}
	}
}

int main() {
	const int n = 10;
	int brojevi[n] = { 1, 2, 3, -6, 0, 0, 12, -2, 856, 711 };
	vector<int> p;
	vector<int> np;

	razvrstaj(brojevi, n, p, np);

	cout << "Parni:" << endl;
	for (int i = 0; i < p.size(); i++) {
		cout << p[i] << " ";
	}
	cout << endl;

	cout << "Neparni:" << endl;
	for (int i = 0; i < np.size(); i++) {
		cout << np[i] << " ";
	}
	cout << endl;

	return 0;
}
```

Zadatak 19

Definirajte strukturu Piksel koja sadrži x koordinatu piksela (od 1 do širine ekrana) i y koordinatu piksela (od 1 do visine ekrana), te vrijednosti između 0 i 255 za crvenu, zelenu i plavu komponentu boje. Napišite program pita korisnika za rezoluciju (širinu i visinu ekrana), a zatim kreira dinamičko polje sa svim pikselima na ekranu tako da rezultirajući ekran bude ovakav:

- Gornja lijeva četvrtina treba biti crvene boje (255 za crvenu, 0 za zelenu i plavu).
- Gornja desna četvrtina treba biti tamno sive boje (50 za crvenu, zelenu i plavu).
- Donja lijeva četvrtina treba biti svjetlo sive boje (150 za crvenu, zelenu i plavu).
- Donja desna četvrtina treba biti plave boje (0 za crvenu i zelenu, 255 za plavu).

Nakon toga, pozivom funkcije iscrtajte ekran tako da crvenu boju iscrtate sa '.', tamno sivu sa '-', svjetlo sivu sa '_', a plavu sa 'o'. Pretpostavite da je ishodište 0, 0 u gornjem lijevom kutu, te da x raste u desnom smjeru, a y prema dolje. Primjerice, ako je širina 60, a visina 16, onda funkcija treba iscrtati:

```
..............................------------------------------
..............................------------------------------
..............................------------------------------
..............................------------------------------
..............................------------------------------
..............................------------------------------
..............................------------------------------
..............................------------------------------
______________________________oooooooooooooooooooooooooooo
______________________________oooooooooooooooooooooooooooo
______________________________oooooooooooooooooooooooooooo
______________________________oooooooooooooooooooooooooooo
______________________________oooooooooooooooooooooooooooo
______________________________oooooooooooooooooooooooooooo
______________________________oooooooooooooooooooooooooooo
______________________________oooooooooooooooooooooooooooo
```

Moguće rješenje:

```
#include <iostream>
#include <vector>
#include <string>
#include <ctime>
using namespace std;

struct Piksel {
	int x;
	int y;
	int r;
	int g;
	int b;
};

void iscrtaj(Piksel ekran[], int n, int sirina) {
	for (int i = 0; i < n; i++) {
		if (ekran[i].r == 255 && ekran[i].g == 0 && ekran[i].b == 0) {
			cout << '.';
		}
		else if (ekran[i].r == 50 && ekran[i].g == 50 && ekran[i].b == 50) {
			cout << '-';
```

```
			}
			else if (ekran[i].r == 150 && ekran[i].g == 150 && ekran[i].b == 150) {
				cout << '_';
			}
			else if (ekran[i].r == 0 && ekran[i].g == 0 && ekran[i].b == 255) {
				cout << 'o';
			}

			// Kad god x postane sirina, znaci da je kraj reda
			if (ekran[i].x == sirina) {
				cout << endl;
			}
		}
}

int main() {
		int sirina;
		cout << "Sirina: ";
		cin >> sirina;

		int visina;
		cout << "Visina: ";
		cin >> visina;

		Piksel* ekran = new Piksel[sirina * visina];
		int indeks = 0;

		for (int redak = 1; redak <= visina; redak++) {
			for (int stupac = 1; stupac <= sirina; stupac++) {
				ekran[indeks].x = stupac;
				ekran[indeks].y = redak;

				if (stupac <= sirina / 2) { // Lijeva strana
					if (redak <= visina / 2) { // Gore
						ekran[indeks].r = 255;
						ekran[indeks].g = 0;
						ekran[indeks].b = 0;
					}
					else { // Dolje
						ekran[indeks].r = 150;
						ekran[indeks].g = 150;
						ekran[indeks].b = 150;
					}
				}
				else { // Desna strana
					if (redak <= visina / 2) { // Gore
						ekran[indeks].r = 50;
						ekran[indeks].g = 50;
						ekran[indeks].b = 50;
					}
					else { // Dolje
						ekran[indeks].r = 0;
						ekran[indeks].g = 0;
						ekran[indeks].b = 255;
					}
				}

				indeks++;
```

```
            }
        }

        iscrtaj(ekran, sirina * visina, sirina);

        delete[] ekran;
        return 0;
}
```

Zadatak 20

Napišite funkciju koja prima dva cjelobrojna parametra i mijenja im vrijednosti. U main-u napravite polje pa pozivajte funkciju za svaka dva susjedna elementa (nulti i prvi, prvi i drugi, drugi i treći, itd.), ali samo ako je lijevi veći od desnog. Ponavljajte prolaze kroz cijelo polje sve dok u jednom prolazu ne napravite 0 zamjena. Na kraju svakog prolaza ispišite polje funkcijom. Primjer rada programa:

```
2 10 9 8 5 1 7 3 4 6
2 9 8 5 1 7 3 4 6 10
2 8 5 1 7 3 4 6 9 10
2 5 1 7 3 4 6 8 9 10
2 1 5 3 4 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
```

Moguće rješenje:

```
#include <iostream>
#include <vector>
#include <string>
#include <ctime>
using namespace std;

void ispisi(int p[], int n) {
	for (int i = 0; i < n; i++) {
		cout << p[i] << " ";
	}
	cout << endl;
}

void zamijeni(int& a, int& b) {
	int temp = a;
	a = b;
	b = temp;
}

int main() {
	const int n = 10;
	int brojevi[n] = { 2, 10, 9, 8, 5, 1, 7, 3, 4, 6 };

	ispisi(brojevi, n);

	bool desila_se_zamjena;
	do {
		desila_se_zamjena = false;
		for (int i = 0; i < n - 1; i++) {
			if (brojevi[i] > brojevi[i + 1]) {
				zamijeni(brojevi[i], brojevi[i + 1]);
				desila_se_zamjena = true;
			}
		}
		ispisi(brojevi, n);
	} while (desila_se_zamjena);

	return 0;
}
```