STRUKTURE PODATAKA I ALGORITMI

Vježbe 04

Zadaci

1. Odaberite konstruktor i kreirajte vektor cijelih brojeva pretpostavljenog (default) kapaciteta. Potom:
   - osigurati kapacitet vektora na vrijednost 100
   - ubaciti u vektor 100 slučajnih brojeva, tako da se svaki od brojeva dodaje na početak, te ispisati brojeve korištenjem *at* pristupa
   - promijeniti veličinu vektora na 50 i korištenjem iteratora ispisati brojeve
   - promijeniti kapacitet vektora na 30 i ispisati brojeve iteratorom unazad
   - kreirati drugi vektor cijelih brojeva kapaciteta 30, te prvi vektor prekopirati u drugi u jednoj liniji
   - očistite sadržaj prvog vektora (svi elementi odjednom)
   - iz drugog vektora izbrišite prvih 10 elemenata te ispišite prvi i posljednji element
   - iz drugog vektora izbrisati preostale elemente jedan po jedan

Zadaci

2. Kreirajte vektor trodimenzionalnih točaka. Iz datoteke „tocke.txt” je potrebno je učitati sve točke u vektor. Nužno je izbjeći kreiranje nepotrebnih objekata.
3. Implementirajte svoj vektor stringova. Vektor ne treba moći rasti već možete koristiti polje veličine 50. Vektor treba moći koristiti na sljedeći način:

```
MojVektor v(10, "HELLOOOO");
for (unsigned i = 0; i < v.size(); i++) {
    cout << v.at(i) << endl;
}
```

4. Ubacite 500 cijelih brojeva u vektor (1 po 1) i ispišite koliko puta je vektor rastao.
5. Izmjerite i ispišite koliko traje ubacivanje 100.000 cijelih brojeva (1 po 1) na početak vektora, a koliko na kraj.

Zadaci

6. Učitajte sve kontakte iz datoteke „kontakti.txt” u vektor. Omogućite korisniku da upiše znak, a nakon toga mu prikažete sve kontakte čije ime ili prezime započinje traženim znakom.
7. Učitajte IP adrese iz datoteke „ip_adrese.txt” u vektor te ispišite sve adrese klase C (vlsm-calc.net/ipclasses.php)
8. Učitajte sve IP adrese iz datoteke „ip_adrese.txt” u vektor te nacrtajte histogram kojim prikazujete koliko ima adrese koje klase, primjerice:

```
Klasa A: ## (2)
Klasa B: # (1)
Klasa C: ### (3)
Klasa D: # (1)
Klasa E:  (0)
```

Zadaci

9. Implementirajte svoj vektor charova. Vektor ne treba moći rasti već možete koristiti polje fiksne veličine 10. Pretpostavimo da će svaki push_back() uspjeti (tj. ne treba provjeravati ima li još mjesta u polju). Vektor treba moći koristiti na sljedeći način:

```
MojVektorChar v;
v.push_back('a');
for (unsigned i = 0; i < v.size(); i++)
    cout << v.at(i) << endl;

v.push_back('b');
for (unsigned i = 0; i < v.size(); i++)
    cout << v.at(i) << endl;

v.push_back('c');
for (unsigned i = 0; i < v.size(); i++)
    cout << v.at(i) << endl;
```

Zadaci

10. Implementirajte svoj vektor stringova koji se može koristiti ovako (omogućite rast primjenom jednostavnog algoritma – ako nema mjesta, alocirajte polje veće za 5 elemenata):

```
void ispisi(MojVektorString& v) {
    cout << "s=" << v.size() << ", c=" << v.capacity() << ": ";
    for (unsigned i = 0; i < v.size(); i++)
        cout << v.at(i);
    cout << endl;
}
int main() {
    MojVektorString v(5, "-");
    ispisi(v);
    for (int i = 0; i < 32; i++) {
        v.push_back("x");
        ispisi(v);
    }
    return 0;
}
```

Zadaci

11.* Implementirajte svoj vektor cijelih brojeva koji se može koristiti ovako (nema rasta, koristite polje fiksne veličine 100):

```
MojVektorNum v;
for (int i = 0; i < 10; i++) {
    v.push_back((i + 1) * 10);
}

for (MojVektorNum::iterator it = v.begin(); it != v.end(); ++it) {
    cout << *it << endl;
}
```