

Egydimenziós tömb (vektor) feltöltése, ha előre ismert a mérete

1. Feltöltés konzolról

```
// lefoglalás
int n;
cout << "méret: ";
cin >> n;
vector<int> t(n);

// feltöltés
for(int i=0; i<n; ++i){ // itt már használható n helyett a t.size()
    cout << "t[" << i << "]= ";
    cin >> t[i];
}
```

2. Feltöltés szöveges állományból, amelyik elsőként a tömb méretét tartalmazza, majd az elemeit elválasztó jelekkel (szóköz, tabulátor, sorvége) tagolva

```
ifstream f("input.txt");
if(f.fail()){
    cout << "Nincs meg a fájl!\n";
    exit(1);
}

// lefoglalás
int n;
f >> n;
vector<int> t(n);

// feltöltés
for(int i=0; i<n; ++i){ // itt már használható n helyett a t.size()
    f >> t[i];
}
```

Egydimenziós tömb (vektor) feltöltése, ha mérete nem ismert előre

1. Feltöltés konzolról

```
vector<int> t;

// feltöltés
int a;
int i = 1;
cout << "t[" << i << "]= ";
cin >> a;
while (!cin.fail()){ // amíg számokat adunk meg
    t.push_back(a);
    ++i;
    cout << "t[" << i << "]= ";
    cin >> a;
}
```

2. Feltöltés szöveges állományból, amelyik nem tartalmazza a tömb méretét, csak az elemeit elválasztó jelekkel (szóköz, tabulátor, sorvége) tagolva

```
ifstream f("input.txt");
if(f.fail()){
    cout << "Nincs meg a fájl!\n";
    exit(1);
}

vector<int> t;

// feltöltés
int a;
while (f >> a){ // amíg számot találunk a fájlban
    t.push_back(a);
}
```

Kétdimenziós tömb (mátrix) feltöltése, amelynek méretét előre kell ismerni

1. Feltöltés konzolról

```
// lefoglalás
int n, m;
cout << "sorok száma: "; cin >> n;
cout << "oszlopok száma: "; cin >> m;
vector<vector<int>> > t;

// feltöltés
t.resize(n);
for(int i=0; i<n; ++i){
    t[i].resize(m);
    for(int j=0; j<m; ++j){
        cout << "t[" << i << "," << j << "]= ";
        cin >> t[i][j];
    }
}
```

2. Feltöltés szöveges állományból, amelyik elsőként a mátrix méreteit tartalmazza, majd az elemeit elválasztó jelekkel (szóköz, tabulátor, sorvége) tagolva sorfolytonos sorrendben

```
ifstream f("input.txt");
if(f.fail()){
    cout << "Nincs meg a fájl!\n";
    exit(1);
}

vector<vector<int>> > t;

// feltöltés
int n, m;
f >> n >> m;

t.resize(n);
for(int i=0; i<n; ++i){
    t[i].resize(m);
    for(int j=0; j<m; ++j){
        f >> t[i][j];
    }
}
```