

## DINAMIČKA MEMORIJA

1. Napisati program u jeziku C++ za nalaženje unije, preseka i razlike dva skupa. Skupovi su zadati kao dinamički nizovi realnih brojeva.

```
1 #include <iostream>
2 using namespace std;
3
4 void ucitaj_niz(double **, int *);
5 void oslobodi_niz(double **);
6 void stampaj_niz(double *a, int n);
7 void unija(double*, int, double*, int, double**, int *);
8 void presek(double*, int, double*, int, double**, int *);
9 void razlika(double*, int, double*, int, double**, int *);
10
11 int main()
12 {
13     int n1, n2, n3;
14     double *a, *b, *c;
15     ucitaj_niz(&a, &n1);
16     ucitaj_niz(&b, &n2);
17
18     unija(a, n1, b, n2, &c, &n3);
19     cout << "Unija skupova: ";
20     stampaj_niz(c, n3);
21     oslobodi_niz(&c);
22
23     presek(a, n1, b, n2, &c, &n3);
24     cout << "Presek skupova: ";
25     stampaj_niz(c, n3);
26     oslobodi_niz(&c);
27
28     razlika(a, n1, b, n2, &c, &n3);
29     cout << "Razlika skupova a/b: ";
30     stampaj_niz(c, n3);
31     oslobodi_niz(&c);
32
33     razlika(b, n2, a, n1, &c, &n3);
34     cout << "Razlika skupova b/a: ";
35     stampaj_niz(c, n3);
36     oslobodi_niz(&c);
37
38     oslobodi_niz(&a);
39     oslobodi_niz(&b);
40     oslobodi_niz(&c);
41     return 0;
42
43     /*
44     * Primeri:
45     * a = {2, 4, 3, 1}, b = {7, 5, 6, 3, 4}
```

```
46 * a = {7, 2, 5, 1, 4}, a = {6, 8, 3}
47 * a = {7, 2, 5, 4}, b = {2, 7, 5, 4}
48 * a = {2, 8, 4, 7, 9}, b = {}
49 * a = {}, b = {}
50 */
51 }
52
53 inline void ucitaj_niz(double **a, int *n)
54 {
55     cin >> *n;
56     *a = new double[*n];
57     for (int i = 0; i < *n; i++)
58     {
59         cin >> (*a)[i];
60     }
61 }
62
63 inline void oslobodi_niz(double **a)
64 {
65     delete [] *a;
66     *a = nullptr;
67 }
68
69 inline void stampaj_niz(double *a, int n)
70 {
71     for (int i = 0; i < n; i++)
72     {
73         cout << a[i] << " ";
74     }
75     cout << endl;
76 }
77
78 inline void unija(double* a, int n1, double* b, int n2, double** c, int *n3)
79 {
80     double *tmp = new double[n1 + n2];
81     for (*n3 = 0; *n3 < n1; (*n3)++)
82     {
83         tmp[*n3] = a[*n3];
84     }
85     for (int i = 0; i < n2; i++)
86     {
87         int j;
88         for (j = 0; j < n1 && a[j] != b[i]; j++);
89         if (j == n1)
90         {
91             tmp[(*)n3++] = b[i];
92         }
93     }
94     *c = new double[*n3];
95     for (int i = 0; i < *n3; i++)
96     {
97         (*c)[i] = tmp[i];
98     }
99     delete [] tmp;
100 }
101
102 inline void presek(double* a, int n1, double* b, int n2, double** c, int *n3)
103 {
104     double *tmp = new double[n1 < n2 ? n1 : n2];
105     *n3 = 0;
```

```
106     for (int i = 0; i < n1; i++)
107     {
108         int j;
109         for (j = 0; j < n2 && a[i] != b[j]; j++);
110         if (j < n2)
111         {
112             tmp[( *n3 )++] = a[i];
113         }
114     }
115     *c = new double[*n3];
116     for (int i = 0; i < *n3; i++)
117     {
118         (*c)[i] = tmp[i];
119     }
120     delete [] tmp;
121 }
122
123 inline void razlika(double* a, int n1, double* b, int n2, double** c, int *n3)
124 {
125     double *tmp = new double[n1];
126     *n3 = 0;
127     for (int i = 0; i < n1; i++)
128     {
129         int j;
130         for (j = 0; j < n2 && a[i] != b[j]; j++);
131         if (j == n2)
132         {
133             tmp[( *n3 )++] = a[i];
134         }
135     }
136     *c = new double[*n3];
137     for (int i = 0; i < *n3; i++)
138     {
139         (*c)[i] = tmp[i];
140     }
141     delete [] tmp;
142 }
```

2. Napisati program u jeziku C++ za računanje transponovane dinamičke matrice realnih brojeva.

```
1 #include <iostream>
2 using namespace std;
3
4 void ucitaj_matricu(double ***, int *, int *);
5 void stampa_matricu(double **, int, int);
6 void oslobodi_matricu(double ***, int, int);
7 void transponovana(double **, int, int, double ***);
8
9 int main()
10 {
11     int n, m;
12     double **a, **b;
13     ucitaj_matricu(&a, &n, &m);
14     transponovana(a, n, m, &b);
15     stampa_matricu(b, m, n);
16     oslobodi_matricu(&a, n, m);
17     oslobodi_matricu(&b, m, n);
18     return 0;
19 }
```

```
20
21 inline void ucitaj_matricu(double ***a, int *n, int *m)
22 {
23     cin >> *n >> *m;
24     *a = new double**[*n];
25     for (int i = 0; i < *n; i++)
26     {
27         (*a)[i] = new double[*m];
28         for (int j = 0; j < *m; j++)
29         {
30             cin >> (*a)[i][j];
31         }
32     }
33 }
34
35 inline void stampaj_matricu(double **a, int n, int m)
36 {
37     for (int i = 0; i < n; i++)
38     {
39         for (int j = 0; j < m; j++)
40         {
41             cout << a[i][j] << " ";
42         }
43         cout << endl;
44     }
45 }
46
47 inline void oslobodi_matricu(double ***a, int n, int m)
48 {
49     for (int i = 0; i < n; i++)
50     {
51         delete [](*a)[i];
52     }
53     delete [](*a);
54     *a = NULL;
55 }
56
57 inline void transponovana(double **a, int n, int m, double ***t)
58 {
59     *t = new double**[m];
60     for (int i = 0; i < m; i++)
61     {
62         (*t)[i] = new double[n];
63     }
64     for (int i = 0; i < n; i++)
65     {
66         for (int j = 0; j < m; j++)
67         {
68             (*t)[j][i] = a[i][j];
69         }
70     }
71 }
```