Prirodno - matematički fakultet
Departman za fiziku

Uvod u objektno - orijentisano programiranje
Školska 2015/2016. godina

DINAMIČKA MEMORIJA

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

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

void ucitaj_niz(double **, int *);
void oslobodi_niz(double **);
void stampaj_niz(double *a, int n);
void unija(double*, int, double*, int, double**, int *);
void presek(double*, int, double*, int, double**, int *);
void razlika(double*, int, double*, int, double**, int *);

int main()
{
   int n1, n2, n3;
   double *a, *b, *c;
   ucitaj_niz(&a, &n1);
   ucitaj_niz(&b, &n2);

   unija(a, n1, b, n2, &c, &n3);
   cout << "Unija skupova: ";
   stampaj_niz(c, n3);
   oslobodi_niz(&c);

   presek(a, n1, b, n2, &c, &n3);
   cout << "Presek skupova: ";
   stampaj_niz(c, n3);
   oslobodi_niz(&c);

   razlika(a, n1, b, n2, &c, &n3);
   cout << "Razlika skupova a/b: ";
   stampaj_niz(c, n3);
   oslobodi_niz(&c);

   razlika(b, n2, a, n1, &c, &n3);
   cout << "Razlika skupova b/a: ";
   stampaj_niz(c, n3);
   oslobodi_niz(&c);

   oslobodi_niz(&a);
   oslobodi_niz(&b);
   oslobodi_niz(&c);
   return 0;

   /*
   * Primeri:
   * a = {2, 4, 3, 1}, b = {7, 5, 6, 3, 4}
   * a = {7, 2, 5, 1, 4}, a = {6, 8, 3}
   * a = {7, 2, 5, 4}, b = {2, 7, 5, 4}
   * a = {2, 8, 4, 7, 9}, b = {}
   * a = {}, b = {}
   */
}

inline void ucitaj_niz(double **a, int *n)
{
   cin >> *n;
   *a = new double[*n];
   for (int i = 0; i < *n; i++)
   {
      cin >> (*a)[i];
   }
}

inline void oslobodi_niz(double **a)
{
   delete[] * a;
   *a = nullptr;
}

inline void stampaj_niz(double *a, int n)
{
   for (int i = 0; i < n; i++)
   {
      cout << a[i] << " ";
   }
   cout << endl;
}

inline void unija(double* a, int n1, double* b, int n2, double** c, int *n3)
{
   double *tmp = new double[n1 + n2];
   for (*n3 = 0; *n3 < n1; (*n3)++)
   {
      tmp[*n3] = a[*n3];
   }
   for (int i = 0; i < n2; i++)
   {
      int j;
      for (j = 0; j < n1 && a[j] != b[i]; j++);
      if (j == n1)
      {
         tmp[(*n3)++] = b[i];
      }
   }
   *c = new double[*n3];
   for (int i = 0; i < *n3; i++)
   {
      (*c)[i] = tmp[i];
   }
   delete[] tmp;
}

inline void presek(double* a, int n1, double* b, int n2, double** c, int *n3)
{
   double *tmp = new double[n1 < n2 ? n1 : n2];
   *n3 = 0;
   for (int i = 0; i < n1; i++)
   {
      int j;
      for (j = 0; j < n2 && a[i] != b[j]; j++);
      if (j < n2)
      {
         tmp[(*n3)++] = a[i];
      }
   }
   *c = new double[*n3];
   for (int i = 0; i < *n3; i++)
   {
      (*c)[i] = tmp[i];
   }
   delete[] tmp;
}

inline void razlika(double* a, int n1, double* b, int n2, double** c, int *n3)
{
   double *tmp = new double[n1];
   *n3 = 0;
   for (int i = 0; i < n1; i++)
   {
      int j;
      for (j = 0; j < n2 && a[i] != b[j]; j++);
      if (j == n2)
      {
         tmp[(*n3)++] = a[i];
      }
   }
   *c = new double[*n3];
   for (int i = 0; i < *n3; i++)
   {
      (*c)[i] = tmp[i];
   }
   delete[] tmp;
}
```

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

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

void ucitaj_matricu(double ***, int *, int *);
void stampaj_matricu(double **, int, int);
void oslobodi_matricu(double ***, int, int);
void transponovana(double **, int, int, double ***);

int main()
{
   int n, m;
   double **a, **b;
   ucitaj_matricu(&a, &n, &m);
   transponovana(a, n, m, &b);
   stampaj_matricu(b, m, n);
   oslobodi_matricu(&a, n, m);
   oslobodi_matricu(&b, m, n);
   return 0;
}
```

```cpp

inline void ucitaj_matricu(double ***a, int *n, int *m)
{
   cin >> *n >> *m;
   *a = new double*[*n];
   for (int i = 0; i < *n; i++)
   {
      (*a)[i] = new double[*m];
      for (int j = 0; j < *m; j++)
      {
         cin >> (*a)[i][j];
      }
   }
}

inline void stampaj_matricu(double **a, int n, int m)
{
   for (int i = 0; i < n; i++)
   {
      for (int j = 0; j < m; j++)
      {
         cout << a[i][j] << " ";
      }
      cout << endl;
   }
}

inline void oslobodi_matricu(double ***a, int n, int m)
{
   for (int i = 0; i < n; i++)
   {
      delete[](*a)[i];
   }
   delete[](*a);
   *a = NULL;
}

inline void transponovana(double **a, int n, int m, double ***t)
{
   *t = new double*[m];
   for (int i = 0; i < m; i++)
   {
      (*t)[i] = new double[n];
   }
   for (int i = 0; i < n; i++)
   {
      for (int j = 0; j < m; j++)
      {
         (*t)[j][i] = a[i][j];
      }
   }
}
```