

```

template <typename T>
void selectSort(fstream& f, T& dummy)
{ int i, j, k;
  T min, a, b;
  // намиране на броя на елементите на f
  f.seekg(0, ios::end);
  long n = f.tellg()/sizeof(T);
  for (i = 0; i < n-1; i++)
  { k = i; f.seekg(k*sizeof(T));
    f.read((char*)&min, sizeof(T));
    for (j = i+1; j < n; j++)
    { f.seekg(j*sizeof(T));
      f.read((char*)&a, sizeof(T));
      if (a < min)
      { k = j;
        min = a;
      }
    }
    f.seekg(i*sizeof(T)); f.read((char*)&a, sizeof(T));
    f.seekg(k*sizeof(T)); f.read((char*)&b, sizeof(T));
    f.seekp(k*sizeof(T)); f.write((const char*)&a, sizeof(T));
    f.seekp(i*sizeof(T)); f.write((const char*)&b, sizeof(T));
  }
  f.close();
}

```


Програма *zad48.cpp* решава задачата. Тя дефинира два шаблона на процедури за създаване и за извеждане на елементите на файл с пряка организация на достъпа. За определяне на типа на елементите на файла се използва параметърът *dummy*. Шаблонът на процедурата *selectSort* е записан във файла *selectSort.cpp*.

```
// Program zad48.cpp
#include <iostream>
#include <fstream>
using namespace std;
#include "selectSort.cpp"

template <typename T>
void readT(T& x)
{ cin >> x;
}

template <typename T>
void createFile(fstream& f, T& dummy)
{ T x;
  readT(x);
  while (cin)
```



```
        { f.write((const char*)&x, sizeof(T));
          readT(x);
        }
    f.close();
}

template <typename T>
void writeT(T x)
{ cout << x << " ";
}

template <typename T>
void writeFile(fstream& f, T& dummy)
{ T x;
  cout << "Sequence\n";
  while (f.read((char*)&x, sizeof(T)))
    writeT(x);
  cout << endl;
  f.close();
}

int main()
{ fstream f("pom.dat", ios::out|ios::binary);
  int dummy;
  createFile(f, dummy);
  f.open("pom.dat", ios::in|ios::out|ios::binary);
  selectSort(f, dummy);
  f.open("pom.dat", ios::in|ios::out|ios::binary);
  writeFile(f, dummy);
  return 0;
}
```