```
#include<iostream>
#include<stdlib.h>
#include <fstream>
#include<strstrea>

int main(int argc, char **argv)
{
   if(argc!=3)
   {
    cout<<"Incorect !\n";
    exit(1);
   }
   ifstream source(argv[1]);
   char line[128];
   if(source.fail())
     cerr<<"Eroare la desch fis"<<argv[1]<<endl;
   else
   {
      ofstream target(argv[2]);
      if(target.fail())
         cerr<<"Eroare la desch fis"<<argv[2]<<endl;
      else
      {
         while(!source.eof())
         {
            source.getline(line, sizeof(line));
            if(source)      //.good())
            {
               target<<line<<endl;
               if(target.fail())
               {
                     cerr<<"Eroare la scriere fis"
                     <<argv[2]<<endl;
                     cin.get();
                     exit(1);
               }
            }
           else if (!source.eof())  // ar putea apare er. la cit.
            //  => dupa citire se pozitioneaza eof
            {
                     cerr<<"Eroare la citirea fis"
                     <<argv[1]<<endl;
                     cin.get();
                     exit(1);
             }
         }
         source.close();            target.close();
      }
   }
  cin.get();   return 0;
}
```

```
#include<iostream>
#include<stdlib.h>
#include <fstream>
class agenda
{
      char nume[80];
      char prefix[4];
      char  numar[6];
    public:
      agenda(){};
      agenda(char *n, char *p, char *nr)
      {
         strcpy(nume,n);
         strcpy(prefix,p);
         strcpy(numar,nr);
      }
     friend ostream& operator<<(ostream& stream,
agenda& a);
      friend istream& operator>>(istream& stream,
agenda& a);
};

ostream& operator<<(ostream& stream, agenda& a)
{
   stream<<a.nume<<" ";    stream<<a.prefix<<" ";
   stream<<a.numar<<"\n";    return stream;
};

istream& operator>>(istream& stream, agenda& a)
{
   cout<<"Nume ";    stream>>a.nume;    cout<<"Prefix ";
   stream>>a.prefix;  cout<<"Numar "; stream>>a.numar;
   cout<<"\n";    return stream;
};

int main()
{
  agenda a;  char c;

  fstream  at("telefon.txt",ios::app);
  if(!at)
  {
     cout<<"Nu pot deschide!!\n";
     return 1;
  }
  for(;;)
  {
     do
     {
        cout<<"1. Introducere\n";     cout<<"2. Afisare\n";
        cout<<"3. Incheiere\n";
        cout<<"Introduceti o optiune: ";          cin>>c;
     }while(c<'1' || c>'3');
     switch(c)
     {
        case '1':
           cin>>a; cout<<"Intrarea este "; cout<<a;
           at<<a;          break;
        case'2':
          char line[255];   at.seekg(0,ios::beg);
          while(!at.eof())
          {
             at.getline(line,sizeof(line),'\n');
             cout<<line<<endl;
          }
          at.clear();//reset eof;
          cout<<endl;            break;
        case '3':
          at.close();
          return 0;
     }
  }

}
```

```
#include <fstream.h>
const int MAX = 100;
int buff[MAX];
void main()
  {
  int j;
  for(j=0; j<MAX; j++)
    buff[j] = j;
  ofstream os;
  os.open("edata.dat",  ios::out |ios::binary);
  if(!os)
  {
     cout<<"Eroare";       cin.get();
  }
  os.write( (char*)buff, MAX*sizeof(int) );
  os.close();
  for(j=0; j<MAX; j++)
    buff[j] = 0;

  ifstream is;
  is.open("edata.dat",  ios::in |ios::binary);

  is.read( (char*)buff, MAX*sizeof(int) );
  for(j=0; j<MAX; j++)
    if( buff[j] != j )
      { cerr << "\nData este incorecta"; return; }
  cout << "\nData este corecta";   cin.get();
}
```

```
#include <fstream.h>
#include <process.h>
const int MAX = 1000;
int buff[MAX];

void main()
  {
  int j;
  for(j=0; j<MAX; j++)    // se umple  buffer-ul cu date
    buff[j] = j;

  ofstream os;
  os.open("edata.dat", ios::out|ios::trunc | ios::binary);
  if(!os)
    { cerr << "\nCould not open output file"; exit(1); }
  cout << "\nWriting...";

  os.write( (char*)buff, MAX*sizeof(int) );
  if(!os)
    { cerr << "\nCould not write to file"; exit(1); }
  os.close();
  for(j=0; j<MAX; j++)
    buff[j] = 0;
  ifstream is;
  is.open("edata.dat", ios::in|ios::binary);
  if(!is)
    { cerr << "\nCould not open input file"; exit(1); }
  cout << "\nReading...";
  is.read( (char*)buff, MAX*sizeof(int) );
  if(!is)
  {cerr << "\nCould not read from file"; exit(1);     }
  for(j=0; j<MAX; j++)                    // check data
    if( buff[j] != j )
      { cerr << "\nData is incorrect"; exit(1); }
  cout << "\nData is correct";   cin.get();
}
```

```
#include <fstream.h>
class person
{
  protected:
    char name[40];
    int age;
  public:
    void getData(void)      {
      cout << "\n   Enter name: "; cin >> name;
      cout << "   Enter age: "; cin >> age;
      }
    void showData(void)
      {
      cout << "\n   Name: " << name;
      cout << "\n   Age: " << age;
      }
 };

void main(void)
{
  char ch;
  person pers;
  fstream file;

  // deschidere pentru append
  file.open("PERSON.DAT", ios::app | ios::out
                              | ios::in | ios::binary );
   do
   {
     cout << "\nEnter person's data:";
     pers.getData();
     file.write( (char*)&pers, sizeof(pers) );
     cout << "Enter another person (y/n)? ";
     cin >> ch;
    }
  while(ch=='y');                  // quit la 'n'
  file.seekg(0);                  // reset la inceputul fisierului
  file.read( (char*)&pers, sizeof(pers) );
  while( !file.fail() )
  {
    cout << "\nPerson:";            pers.showData();
    file.read( (char*)&pers, sizeof(pers) );
  }
  ifstream infile;  // creare urmata de deschidere bin
  infile.open("PERSON.DAT", ios::in|ios::binary);
  infile.seekg(0, ios::end);          //  0 bytes fata de end
  int endposition = infile.tellg();        // pozitionare la end
  int n = endposition / sizeof(person);    // nr de pers.
  cout << "\nSunt " << n << " persoane in fisier";
  cout << "\nIntroduceti numarul persoanei: ";
  cin >> n;
  int position = (n-1) * sizeof(person);
  infile.seekg(position);   // plasare pe inregistrarea dorita
  infile.read( (char*)&pers, sizeof(pers) );
  pers.showData();
  cin.ignore(10,'\n');
  cin.get();
}
```