Automaton minimization II.

Submission deadline:	**2011-11-19 23:59:59**	2764675.509 sec
Evaluation:	**0.0000**	
Max. assessment:	**3.0000** (Without bonus points)	
Submissions:	0 / 20	
Advices:	0 / 0	

Second task

Your task is to implement the automaton from the first part of the homework

Implement the minimized automaton from the first task in the form of a C++ function, whose signature is as follows:

```
/**
 * students' interface
 * @param inputString null terminated string with automaton input
 * @return true if string has been accepted
 */
bool checkString(const char * inputString);
```

The input is a string of 8-bit characters of random length, terminated by a '\0' character. The output of the function is value 'true' if and only if the automaton accepts the string (otherwise, output is 'false').

You shall submit one compilable C++ source file that will be processed using the same mechanisms as tasks from courses PA1/PA2. It is thus recommended to wrap conflicting parts of code (mainly platform-dependent code and possible naming conflicts) in a conditional compilation macro based on the identifier __PROGTEST__. For example:

```
#ifndef __PROGTEST__

int main() {
    using namespace std;
    string line;
    while (getline(cin, line)){
        cout << checkString(line.c_str()) << endl;
    }
}

#endif
```

Try to name your functions and variables using unique identifiers where there is no chance of conflict with functions from the standard library. You can prefix them with `aag_`. If you follow this advice, you cannot run into naming conflicts during progtest compilation. Submit your file in plain ASCII encoding, please.

Submit:	[redacted]	**Submit**