```cpp
/*--- HWSol.cpp ---*/
//Mike Hanling

#include <iostream>
#include <fstream>
#include <string>
#include <vector>
using namespace std;

class dfa {
  //states will be sequential integers from 0 (0 will be initial state)
  //accepting states is list of len q will a true/false for each state
  public:
    int state = 0;
    vector<vector<int>> d;
    vector<bool> w;
    void transition(char x) {
      int z = x - 'a';
      state = d[state][z];
    }
    bool isAccepting() {
      return w[state];
    }
};

istream& operator>> (istream& in, dfa& m) {
  string junk;
  int val;
  vector<int> vals;
  int row = 0;
  int col = 0;

  in >> row >> junk >> col;
  for (int i = 0; i < row; ++i) {
    vals.clear();
    for (int j = 0; j < col; ++j) {
      in >> val;
      vals.push_back(val);
    }
    m.d.push_back(vals);
  }

  for (int i = 0; i < row; ++i) {
    int val;
    in >> val;
    m.w.push_back(val == 1 ? true : false);
  }

  return in;
}


int main() {
  //ask user for input file of FA
  cout << "What is the name of the file describing the FA?" << endl;

  //read in from file
  string filename;
  cin >> filename;
  ifstream fin(filename);


  //make an instance of fa and fill it with read info
  dfa m;
  fin >> m;

  //read in string from user
  string in;
  cin >> in;

  //step through every letter and change the current state
  for (int i=0; i < in.length(); ++i) {
    m.transition(in.at(i));
  }

  //output if the ending state is accepting or not
  cout << (m.isAccepting() ? "accept" : "reject") << endl;

  return 0;
}
```