CENG 242

Hw #2

Spring 2006/2007

(Due: April 1st, 2007 Sunday 23:59)

In this homework, you will write a Haskell code determining whether a given propositional formula is satisfiable or not. A propositional formula is defined as:

```
data Formula      = Atomic Char
                  | Disjunction Formula Formula
                  | Conjunction Formula Formula
                  | Negation Formula
```

e.g. `(Disjunction (Conjunction (Atomic 'p') (Atomic 'q')) (Negation (Atomic 'p')))` means $((p \wedge q) \vee \neg p)$ mathematically.

You will write a function **satisfy** in the form:

```
satisfy :: Formula -> [Char]
```

The function should get a propositional formula and return a list of atoms, such that, when you assign True to these atoms and False to remaining atoms, the formula evaluates to True. If the formula is unsatisfiable, then return the string **"!"**.

Specifications:

- The characters that describe atoms will be lower case characters ['a'..'z'].
- For the definitions of disjunction, conjunction etc. you can use www.wikipedia.org.
- There may be more than one solution. Any correct solution will be considered as OK.
- You will submit a single file named Hw2.hs including all your definitions. There must be a module called Hw2. So your file (Hw2.hs) should begin with a line:

  module Hw2 where
  ....

- You will submit your codes through cow system. Specifications (file name, function name, module name etc.) are strict. Breaking any of them will cost you getting a 0 from the homework since black box method is used.

Examples:

```
> satisfy (Atomic 'p')
"p"
```

```
> satisfy (Conjunction (Atomic ‘p’) (Atomic ‘r’))
“pr” –- “rp” is also OK

> satisfy (Conjunction (Atomic ‘r’) (Negation (Atomic ‘s’)))
“r”

> satisfy (Disjunction (Atomic ‘p’) (Atomic ‘t’))
“p” –- “t”, “pt” and “tp” are also OK

> satisfy (Conjunction (Atomic ‘p’) (Negation (Atomic ‘p’)))
“!”
```