Haskell — Various factorials P32333_en

Write 8 *essentially diferent* functions that compute the factorial of a natural number in Haskell. You can use recursion, final recursion, no recursion, guards, **if then else**, `map`, `fold`s, infinite lists...

Specification

You should define the following functions:

fact1 :: **Integer** → **Integer**
fact2 :: **Integer** → **Integer**
fact3 :: **Integer** → **Integer**
fact4 :: **Integer** → **Integer**
fact5 :: **Integer** → **Integer**
fact6 :: **Integer** → **Integer**
fact7 :: **Integer** → **Integer**
fact8 :: **Integer** → **Integer**

Sample input

```
fact1 6
fact2 6
fact3 6
fact4 6
fact5 6
fact6 6
fact7 6
fact8 6
fact1 20
```

Sample output

```
720
720
720
720
720
720
720
720
2432902008176640000
```

Problem information

Author : Jordi Petit
Translator : Jordi Petit
Generation : 2024-04-30 19:14:54