

函数程序设计实验二(2)

设计一个显示放大字符串的程序

编写一个显示放大字符串的程序。假设字符串由字母 (a,b,...,z) 和数字(0,1,...,9)组成, 字母不分大小写。设计下列函数:

```
say :: String -> String
sayit :: String -> IO ()
sayit = putStr . say
```

例如, 运行

```
>sayit "Hello"
```

屏幕得到显示结果

```
H  H  EEEEE L    L    000
H  H  E    L    L    0  0
HHHHH EEEEE L    L    0  0
H  H  E    L    L    0  0
H  H  EEEEE LLLL  LLLL  000
```

再如, 运行

```
>sayit "Hi 123"
```

屏幕得到显示结果

```
H  H  IIIII          1    222  3333
H  H   I            11   2  2    3
HHHHH   I           1    2    333
H  H   I            1    2    3
H  H  IIIII        11111 22222 3333
```

注: 你可能需要使用模块Char或者List函数, 例如

```
isUpper :: Char -> Bool
isLower :: Char -> Bool
ord      :: Char -> Int
chr      :: Int  -> Char
unlines  :: [String] -> String
```

查看模块Char和List或者在Hugs中调入模块以了解这些函数的含义。在ghci中调入模块, 如Char的方法:

- 运行ghci, 运行import命令:

```
>import Data.Char
```

- 运行Hugs或者Winhugs, 然后利用命令:l 模块名调入模块, 如

```
>:l Char
```

- 试着查看函数的类型和应用效果, 例如

```
>:t ord
ord :: Char -> Int
> ord 'a'
97
> ord 'A'
65
> :t chr
chr :: Int -> Char
> chr 65
'A'
```

递交实验要求

- 你提交的模块和文件均命名为MyPicture, 如模块形如

```
module MyPicture where
import Data.Char
import Data.List

triads :: Int -> [(Int, Int, Int)]
-- triads 的定义
sayit :: String -> IO ()
```

```
sayit = putStr . say
```

```
say :: String -> String
```

```
-- say 的定义
```

- 通过Elearning.sysu.edu.cn提交。
- 如果你的程序尚不能运行，请用注释的方式说明理由或者困难。
- 实验记入成绩，请认真对待。
- 切勿抄袭，后果严重。