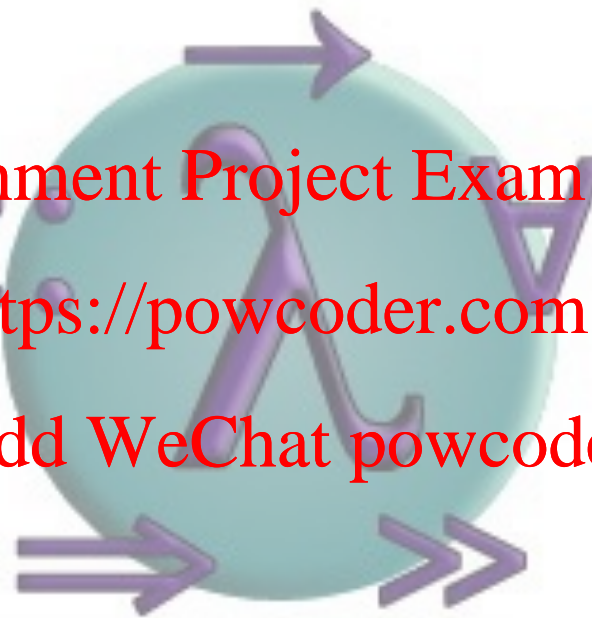


PROGRAMMING IN HASKELL

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Chapter 9 - Interactive Programs

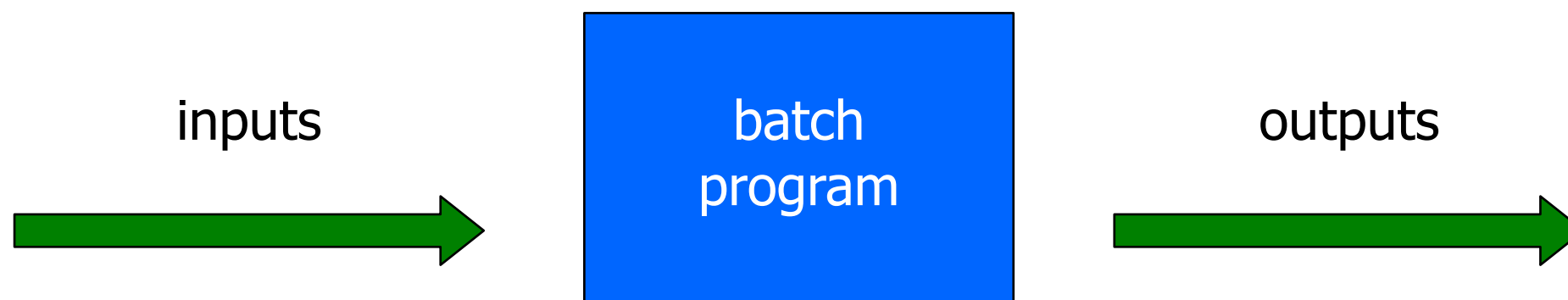
Introduction

To date, we have seen how Haskell can be used to write batch programs that take all their inputs at the start and give all their outputs at the end.

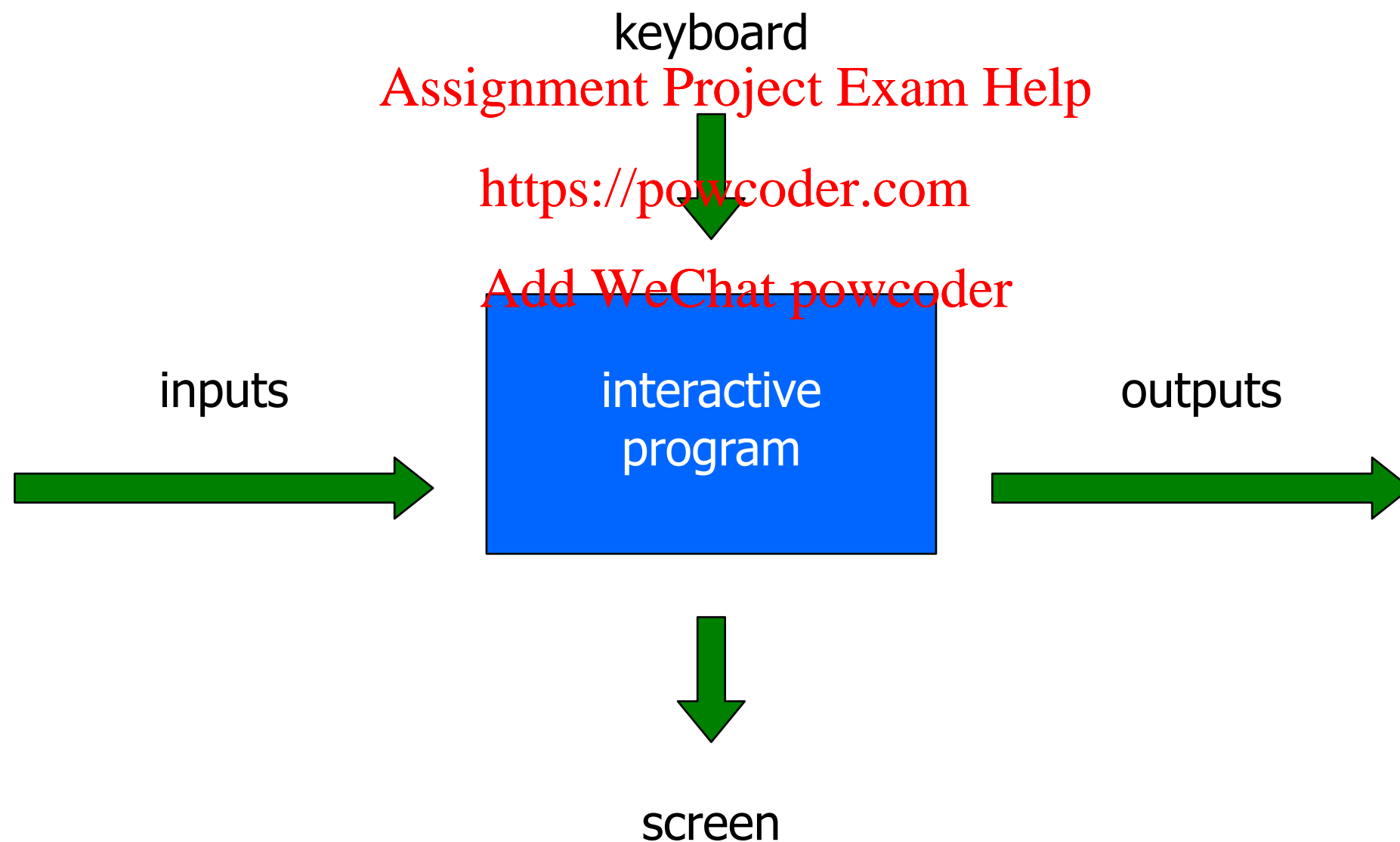
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



However, we would also like to use Haskell to write interactive programs that read from the keyboard and write to the screen, as they are running.



The Problem

Haskell programs are pure mathematical functions:

 Haskell programs have no side effects.

<https://powcoder.com>

Add WeChat powcoder

However, reading from the keyboard and writing to the screen are side effects:

 Interactive programs have side effects.

The Solution

Interactive programs can be written in Haskell by using types to distinguish pure expressions from impure actions that may involve side effects.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

IO a

The type of actions that return a value of type a.

For example:

IO Char

The type of actions that return a character.

Assignment Project Exam Help

<https://powcoder.com>

IO ()

The type of purely side effecting actions that return no result value.


Note:

 () is the type of tuples with no components.

Basic Actions

The standard library provides a number of actions, including the following three primitives:

Assignment Project Exam Help

-  The action getChar reads a character from the keyboard, echoes it to the screen, and returns the character as its result value:

<https://powcoder.com>

Add WeChat powcoder

```
getChar :: IO Char
```

- ? The action putChar c writes the character c to the screen, and returns no result value:

`putChar :: Char → IO ()`

Assignment Project Exam Help

<https://powcoder.com>

- ? The action return v simply returns the value v, without performing any interaction:

`return :: a → IO a`

Sequencing

A sequence of actions can be combined as a single composite action using the keyword do.

For example:

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
act :: IO (Char,Char)
act = do x ← getChar
        getChar
        y ← getChar
        return (x,y)
```

do-notation again

IO also supports the do-notation and again, similarly to what happened with Parsers, we need to be careful understanding the typing of the \leftarrow

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

getChar :: IO Char

```
act :: IO (Char,Char)
act = do x ← getChar
        char
        getChar
        return (x,y)
```

x :: Char

Derived Primitives

 Reading a string from the keyboard:

Assignment Project Exam Help
<https://powcoder.com>
Add WeChat powcoder

```
getLine :: IO String
getLine = do x ← getChar
  if x == '\n' then
    return []
  else
    do xs ← getLine
      return (x:xs)
```

? Writing a string to the screen:

```
putStr    :: String → IO ()
```

```
putStr [] = ?
```

```
putStr (x:xs) = ?
```

<https://powcoder.com>

Add WeChat powcoder

? Writing a string and moving to a new line:

```
putStrLn  :: String → IO ()
```

```
putStrLn xs = do putStr xs
```

```
putChar '\n'
```

Example

We can now define an action that prompts for a string to be entered and displays its length:

Assignment Project Exam Help

```
strlen :: IO ()  
strlen = do putStrLn "Enter a string: "  
           xs ← getLine  
           putStrLn "The string has "  
           putStrLn (show (length xs))  
           putStrLn " characters"
```

<https://powcoder.com>

Add WeChat powcoder

For example:

```
> strlen
```

```
Enter a string: abcde
```

```
The string has 5 characters
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Note:

-  Evaluating an action executes its side effects, with the final result value being discarded.

Hangman

Consider the following version of hangman:

- ❑ One player secretly types in a word.
- ❑ The other player tries to deduce the word, by entering a sequence of guesses.
- ❑ For each guess, the computer indicates which letters in the secret word occur in the guess.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

 The game ends when the guess is correct.

We adopt a top down approach to implementing hangman in Haskell, starting as follows:

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
hangman :: IO ()
hangman =
  do putStrLn "Think of a word: "
     word ← sgetLine
     putStrLn "Try to guess it:"
     play word
```


The action `getCh` reads a single character from the keyboard, without echoing it to the screen:

```
import System.IO
```

Assignment Project Exam Help

```
getCh :: IO Char
```

<https://powcoder.com>

```
getCh = do hSetEcho stdin False
```

Add WeChat powcoder

```
  x ← getChar
```

```
  hSetEcho stdin True
```

```
  return x
```

The action sgetline reads a line of text from the keyboard, echoing each character as a dash:

Assignment Project Exam Help

<https://powcoder.com>

sgetline :: IO String
sgeline = ?

Add WeChat powcoder

The function match indicates which characters in one string occur in a second string:

```
match :: String → String → String
```

```
match xs ys = ?
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

For example:

```
> match "haskell" "pascal"
```

```
"-as--ll"
```

The function `play` is the main loop, which requests and processes guesses until the game ends.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
play :: String → IO ()  
play word =  
  ?
```

Exercise

Implement the game of nim in Haskell, where the rules of the game are as follows:

Assignment Project Exam Help

? The board comprises five rows of stars: <https://powcoder.com>

Add WeChat powcoder

```
1: * * * * *
2: * * * *
3: * * *
4: * *
5: *
```

- ❓ Two players take it turn about to remove one or more stars from the end of a single row.
- ❓ The winner is the player who removes the last star or stars from the board.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Hint:

Represent the board as a list of five integers that give the number of stars remaining on each row. For example, the initial board is [5,4,3,2,1].