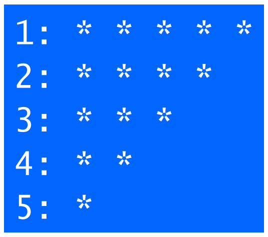
Assignment No. 9
EECS 468
Programming Language Paradigms
Due: 11:59 PM, Wednesday, December 6, 2023
Submit deliverables in a single zip file to Canvas
Files in other formats (e.g., .tar) will not be graded
Name of the zip file: FirstnameLastname_Assignment9 (with your first and last name)
Name of the Assignment folder within the zip file: FirstnameLastname_Assignment9

Deliverables:

1. Copy of Rubric9.docx with your name and ID filled out (do not submit a PDF)
2. Haskell script file(s).
3. Screen print of playing nim one time. It doesn't matter which player wins. (Copy and paste the output to a Word document and PDF it).

Assignment:

- Implement the game of <u>nim</u> in Haskell, where the rules of the game are as follows:
    - The initial board comprises five rows of stars:



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

    - Two players take turns removing 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.
- Define:

  type Board = [Int]
  initial :: Board
  initial = [5,4,3,2,1]

- Implement a recursive main game loop which takes the current board and player number as arguments: play :: Board -> Int -> IO()
- The main game loop should:
    - First display the board.
    - Then check if the game is finished
    - If so, then display the other player as the winner as they are the one who made the board empty.
    - Otherwise, prompt the current player for the move they wish to make with the following prompts:

      Player – followed by the player number
      Enter a row number:
      Stars to remove:

    - If the move is valid, update the board accordingly, display it, and then continue the game with next player.

  - Otherwise display an error message, re-display the board, and re-prompt the current player to enter a valid move.
  - Each board display, player prompts display, and winner display should be proceeded by a newline. For example:

```
> nim

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

Player 1
Enter a row number: 1
Stars to remove: 4

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

Player 2
Enter a row number: 6
Stars to remove: 1
ERROR: Invalid move (This can be any error message, not this one
specifically)

Etc.
```

- Invoke the game nim itself by invoking the game loop with the initial board and player number:

```
nim :: IO ()
nim = play initial 1
```

- Hint: do all the IO in the main "play" function; segregate purely IO functions (e.g., putChar) from purely functional functions (e.g., check validity of board).
- You might find these two imports useful:
  - import Data.Char (isDigit)
  - import Data.Char (digitToInt)
- Provide comments for the Haskell code that explain what each line of code is doing. See rubric below.

Rubric for Program Comments		
Exceeds Expectations (90-100%)	**Meets Expectations (80-89%)**	**Unsatisfactory (0-79%)**
Software is adequately commented with prologue comments, comments summarizing major blocks of code, and comments on every line.	Prologue comments are present but missing some items or some major blocks of code are not commented or there are inadequate comments on each line.	Prologue comments are missing all together or there are no comments on major blocks of code or there are very few comments on each line.

Adequate Prologue Comments:

- Name of program contained in the file (e.g., EECS 468 Assignment 7 - Replicate)
- Brief description of the program, e.g.:
    - Haskell function for replicate
- Inputs,e.g.,:
    - Number of replications
    - Element to replicate
- Output, e.g.,
    - List of replicated elements
- All collaborators
- Other sources for the code ChatGPT, stackOverflow, etc.
- Author's full name
- Creation date: The date you first create the file, i.e., the date you write this comment

Adequate comments summarizing major blocks of code and comments on every line:

- Provide comments that explain what each line of code is doing.
- You may comment each line of code (e.g., using --) and/or provide a multi-line comment (e.g., using {- and -}) that explains what a group of lines does.
- Multi-line comments should be detailed enough that it is clear what each line of code is doing.
- Each block of code must indicate whether you authored the code, you obtained it from one of the sources listed in the prolog, or one of your collaborators authored the code, or if it was a combination of all of these.

Collaboration and other sources for code:

- When you collaborate with other students or use other sources for the code (e.g., ChatGPT, stackOverflow):
    - Your comments must be significantly different from your collaborators.
    - More scrutiny will be applied to grading your comments in particular explaining the code "in your own words", not the source's comments (e.g., ChatGPT's comments).
- Failure to identify collaborators or other sources of code will not only result in a 0 on the assignment but will be considered an act of Academic Misconduct.

- Students who violate conduct policies will be subject to severe penalties, up through and including dismissal from the School of Engineering.