

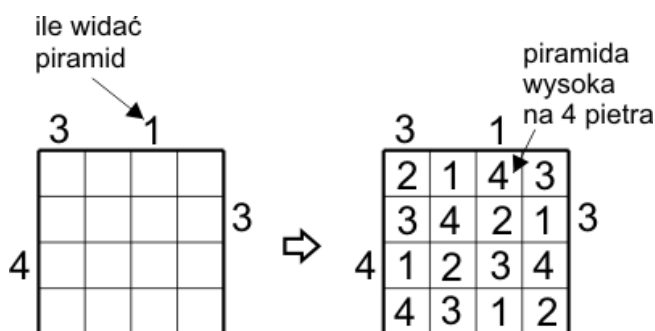


## Zadanie projektowe z języka Haskell

Semestr: lato 2016

### Opis zadania

Zadanie realizowane jest w dwuosobowych zespołach. Skład zespołu należy podać przy oddawaniu rozwiązania (patrz niżej). W ramach zadania należy napisać aplikację rozwiązującą następującą łamigłówkę. Dana jest plansza o wymiarze  $n$  na  $n$ . Na planszy należy ustawić piramidy o wysokościach od 1 do  $n$ . W każdym wierszu i kolumnie należy ustawić piramidy różnej wysokości. Przy niektórych kolumnach i wierszach umieszczone są liczby o wartości od 1 do  $n$ , które oznaczają ile piramid widać w rzędzie z tego miejsca (wyższe piramidy zasłaniają niższe). Na przykład, wartość 1 oznacza, że widać tylko jedną, najwyższą piramidę (zatem musi być ona pierwsza w rzędzie); a wartość  $n$ , że widać wszystkie  $n$  piramid (zatem muszą być one ustawione od najniższej do najwyższej). Każda łamigłówka ma dokładnie jedno rozwiązanie. Poniżej znajduje się przykładowa łamigłówka z rozwiązaniem:



Powyższy przykład i inne można znaleźć pod adresem: [www.wydawnictwologi.pl/piramida](http://www.wydawnictwologi.pl/piramida).

Algorytm poszukiwania rozwiązania łamigłówki jest częścią zadania. Po uruchomieniu aplikacja powinna spytać użytkownika o nazwę pliku wejściowego z łamigłówką. Taki plik zawiera dokładnie jeden wiersz postaci:

```
Piramidy [wartości miejsc nad planszą] [wartości miejsc pod planszą] [wartości
miejsc z lewej strony planszy], [wartości miejsc z prawej strony planszy]
```

Wartości miejsc wokół planszy oddzielone są przecinkami i mają postać „Just liczba”, jeżeli w danym miejscu jest liczba, lub „Nothing” jeżeli w danym miejscu nie ma liczby. Plik wejściowy dla pokazanej wyżej łamigłówki ma postać:

```
Piramidy [Just 3, Nothing, Just 1, Nothing] [Nothing, Nothing, Nothing,
→ Nothing] [Nothing, Nothing, Just 4, Nothing] [Nothing, Just 3, Nothing,
→ Nothing]
```

Można założyć poprawność wczytywanego pliku. Nie ma ograniczeń na wielkość planszy. Po znalezieniu rozwiązania, aplikacja powinna je wyświetlić w formie tekstowej na ekranie.

## Termin i forma oddania zadania

Źródła programu należy wysłać w formie załącznika na adres [m.szlenk@elka.pw.edu.pl](mailto:m.szlenk@elka.pw.edu.pl) do dnia 30 maja 2016 r. (włącznie). Mail powinien być zatytułowany „[SPOP Haskell] Nazwisko1 Nazwisko2”. Dodatkowo należy również dołączyć dokument (.pdf, .rtf lub .doc) zawierający jedną (i tylko jedną) stronę opisu do źródeł. Opis ten może mieć charakter półformalny (odpowiednik osobistego prezentowania źródeł prowadzącemu). Należy w nim założyć, że czytelnik wie do czego program służy, natomiast chce zorientować się, co i gdzie jest w kodzie i jaka jest idea rozwiązania. Poziom szczegółowości opisu wyznacza jego maksymalny rozmiar równy jednej stronie.

## Kryteria oceny

Projekt jest oceniany w skali 0-20 punktów. Przy ocenie będą brane pod uwagę:

- zakres funkcjonalny programu (czy program realizuje wymienione w opisie zadania funkcje?),
- zachowanie programu (czy jest zgodne z oczekiwaniami?),
- czytelność i intuicyjność interfejsu,
- zwięzłość i czytelność kodu, w tym jakość komentarzy,
- przydatność (w trakcie dokonywania przeglądu kodu) dołączonej „dokumentacji”.