CA: TRABAJO PRÁCTICO

Seleccionar un lenguaje de programación, y escribir en Haskell un **pretty-printer** para ese lenguaje, es decir una función que permita imprimir código embellecido, de modo que se genere un tabulado en función del anidamiento de bloques dentro del código, y que a la vez las sentencias (de haberlas) queden en líneas separadas.

La función principal debe recibir una lista de Char, que representa el código de un programa en el lenguaje elegido, y una lista de enteros que representa las posiciones de tabulado a seguir, y deberá devolver como salida una lista de Char que represente al código del programa con los espacios insertados de acuerdo con el anidamiento de bloques y a las cantidades de espacios por cada tabulado en secuencia (definida por el segundo parámetro).

```
funcionPrincipal :: ([[Char]], [Int]) -> [Char]
```

Si el programa ingresado ya tuviese tabulados o espacios, queda opcionalmente la posibilidad de que esta función los corrija de modo que la salida respete el tabulado deseado. Para lenguajes que manejen sentencias (por ej. en C están separadas por ;), en la salida las mismas deberán quedar en líneas separadas.

El lenguaje objeto elegido debe contemplar el uso de bloques, como Algol, Pascal, C, Java, Javascript, Smalltalk, etc. Es necesario que el tabulado producido respete la secuencia de espacios ingresada. C, Java y Javascript para delimitar bloques usan { y }. Smalltalk usa [y]. Pascal y ADA usan begin y end. Visual Basic usa variados bloques, como: If ... End If, Select Case... End Select y otros. BF usa [y].

Si la lista de enteros ingresada no fuese suficientemente larga para el nivel de anidamiento dado por el código, se puede adoptar como criterio utilizar el último entero de ésta para los tabulados siguientes, o bien un recomienzo de los valores en dicha lista (y así sucesivamente), o bien arrojar error. La decisión sobre cualquier caso especial o no contemplado queda como parte de la resolución.

Por supuesto, tanto en la entrada como en la salida podría haber caracteres especiales tales como '\n' (salto de línea). En la salida los '\n' son necesarios para separar líneas.

Ejemplo para C:

Si ingresa el código:

```
void main(){int p=i=1;while(i<=10){p*=i++;}printf("%d\n",p);}
```

y la lista de enteros [2,4,1]

La salida será algo como:

```
void main(){
  int p=i=1;
  while(i<=10){
     p*=i++;
  }
  printf("%d\n",p);
}
```

De ser necesario, después ponemos o vemos otros ejemplos reales.

Integrantes por equipo: de 1 a 3 (consultar por cualquier duda al respecto).

Entrega: por e-mail en un archivo Ascii (.txt o .hs) con el código Haskell (todas las funciones usadas, con declaraciones de tipo y cuerpos, que se lean y anden correctamente en la máquina); y en el mismo archivo u otro (.txt, .doc o .pdf) al menos 4 ejemplos de uso. Se pueden usar funciones auxiliares, resueltas en las clases y/o resueltas de la guía. Todo funcional puro, con los comentarios necesarios y siguiendo los usos y estilos vistos en las clases.

Fecha: hasta el miércoles 12/6.

Llegado el caso podría ser necesaria una breve evaluación oral individual sobre el contenido de la entrega (en especial las técnicas usadas), durante la misma o algún día posterior.