Assignment #3
on Scheme and Common Lisp

Date Due: March 18, 2026

Total: 100 marks

Include for all programs both rules/code and execution (.scm, or .lis souce code, a README and a RUN file for each program).

Include for all programs both rules/code and execution. Don't use code from the internet. Write your own code. Don't use built-in functions that will solve the problem for you; that will defend the purpose of the assignment.

Write the corresponding programs in both languages Common Lisp and Scheme for the following:

1. (30 marks) Write a function to interactively guess if a number has sum property in relation to a list stored internaly in at most k tries. The number k is read from standard input. After k is entered, the user must enter a number x and k values i_1, i_2, i_k, $1 \leq i_1, i_2, \ldots, i_k \leq n$.

   A number x has sum property with respect to the list $[A_1, \ldots A_n]$ if the indexes satisfy this condition $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ and $x = \sum_{i=0}^{k} A_{i_k}$.

   If the number has sum property, the programs ends(user wins), if not, the program displays the number of tries left and the user tries to guess the number again.

   In case the user types a number that is lower than the sum of all numbers in the list this should be displayed. After each try, the number of remaining tries should be displayed.

   If the numbers i_1, i_2, i_k, $1 \leq i_1, i_2, \ldots, i_k \leq n$ entered by the user do not have the property that $1 \leq i_1 < i_2 < \ldots < i_k \leq n$, the program decreses the number of tries by two and the user must guess again.

   Outputs can be obtained with function `(write X)` in clisp and `(display X)`, in scheme, and for an input we can use the function `(read X)` (in both Scheme and Lisp). The list of numbers should be hardcoded (inside the program).

   For example:

```
(game)
6
You have 6 tries to guess a number with sum property.
60
1
2
3
4
5
6
```

```
The max sum is lower.
You have 5 tries to guess a number with sum property.
20
1
3
5
7
4
You have 3 tries to guess a number with sum property.
10
1
4
6
You guessed right.
```

2. (30 marks) Write a function to interactively mix two lists (stored internally) into one list depending on an the value of an integer k is read from standard input. The first k elements are obtained as follows:

   (a) The first element of the list is the maximum of the first element of the first list and the k-th element of the second list.

   (b) The second element of the list is the minimum of the second element of the first list and the $k-1$-th element of the second list.

   (c) the rest of the elements are obtained by alternatively applying the previous two rules.

   If k is greater than the minimum of the lengths of both lists the user must input the number again.

   The next elements are obtained by mixing the rest of the elements, one from the second list one from the first list, starting with $(k+1)$th elements. If one list is longer then the other one, the missing elements from the shorter list are ignored.

   For example:

```
(game)
Number k:4
Initial Lists: [1,4,5,7,9,10,-3,2,8,9][7,14,1,-7,-9,10,3,-2]
Result [1,1,14,7,9,-9,10,10,-3,3,2,-2,8,9]
```

```
(game)
Number k:12
The number is too big enter a lower number
11
The number is too big enter a lower number
3
Initial Lists: [1,4,5,7,9,10,-3,2,8,9][7,14,1,-7,-9,10,3,-2]
Result [1,4,7,7,-7,9,-9,10,10,-3,3,2,-2,8,9]
```

```
(game)
Number k:12
The number is too big enter a lower number 11
The number is too big enter a lower number 3
Initial Lists: [11,4,5,7,9][7,14,1,-7,-9,10,3,-2]
Result [11,4,7,7,-7,9,-9,10,3,-2]
```

Newlines or other extra whitespaces are optional – they are added just to allow us to read.