

## נספח ב- PROLOG

### ניתן להשתמש בכל אחד מהפרדיקטים הבאים ואין צורך להגדירם

append(L1,L2,Result).	משרשר את L1 ו-L2 לתוך Result
delete(X,L,Result).	מוחק את X מתוך L והתוצאה ב-Result
last(X,List).	בודק האם X הוא האיבר האחרון מתוך List (או מחזיר ב-X את האיבר האחרון כנ"ל).
length(L,N).	מחזיר ב-N את מספר האיברים ברמה העליונה ב-L.
member(X,List).	בודק אם X הוא איבר ב-List (או מחזיר איבר כנ"ל, כל פעם איבר אחר).
permutation(OriginalList,Permutation).	מחזיר פרמוטציות של OriginalList
prefix(Prefix,List).	בודק אם Prefix הוא תחילית של List (או מחזיר תחילית כנ"ל).
subList(S,L).	בודק האם S היא תת רשימה של L (או מחזיר תת רשימה של L).
sum_list(L,S).	כאשר L היא רשימת מספרים, S תכיל את סכום איברי L.
max_list(L,S).	כאשר L היא רשימת מספרים, S תכיל את המקסימום מבין איברי L.
min_list(L,S).	כאשר L היא רשימת מספרים, S תכיל את המינימום מבין איברי L.
atom(X)	בודק האם X הוא אטום
integer(X)	בודק האם X הוא מספר שלם.
number(X)	בודק האם X הוא מספר.
atomic(X)	בודק האם X הוא אטום או מספר
var(X)	בודק האם X הוא משתנה לא מאותחל
nonvar(X)	בודק האם X אינו משתנה לא מאותחל
is_list(X)	בודק האם X הוא רשימה
write(X)	מדפיס את X
nl	עובר שורה בהדפסה

## נספח ב-Python

<u>Built-in data types</u>	
<b>immutable data types</b> : int, float, str, tuple <b>mutable data types</b> : list, set, dict <b>iterables</b> : str, tuple, list, set, dict <b>data type conversion functions</b> : int(), float(), str(), tuple(), list(), set(), dict()	<b>sequences</b> : str, tuple, list <b>item retrieval</b> : $seq[idx] \rightarrow itemValue$ - $0 \leq idx < len(seq)$ or $-len(seq) \leq idx \leq -1$ <b>slice</b> : $seq[i:j:k]$ - sub-sequence determined by i, j-1, and k (all optional) $seq.index(itemValue) \rightarrow itemIdx$ <b>reversed</b> (seq) $\rightarrow reversedSeq$ - creates a new sequence in which all the top-level items of seq occur in reverse order. <b>sorted</b> (seq, reverse=[True False]) - creates a new sequence in which all the top-level items of seq occur sorted in ascending order (if reverse=True, it is in descending order). <b>zip</b> ( $L_1, \dots, L_N$ ) $\rightarrow [(L_{11}, L_{21}, \dots, L_{N1}), \dots, (L_{1M}, L_{2M}, \dots, L_{NM})]$ where N is number of lists passed to zip and M is the length of all those N lists.
<b>str</b> - given a string S: S.find(substr) $\rightarrow idx$ (-1 if not found) S.count(substr) $\rightarrow$ nr of substr occurrences S.split(sep) $\rightarrow$ list of substrings separated by sep in S.	<b>list</b> - given a list L: L[idx] = value - update of L[idx] with value. L.append(value) - appends value to the end of the list L. L.reverse() - reverses the order of the top-level items of a list L, in place. L.sort(reverse = [True False]) - sorts the list L in ascending order, in place. ...
<b>dict</b> - given a dictionary D: <b>item retrieval</b> : D[key] $\rightarrow$ value <b>create/update item</b> : D[key] = value D.items() $\rightarrow$ pairs (key,value) iterator D.keys() $\rightarrow$ keys iterator D.values() $\rightarrow$ values iterator	
<b>isinstance(value, type) or isinstance(value, (type<sub>1</sub>, ..., type<sub>n</sub>))</b> - boolean function that tests if value is an instance of the type data type.	
<b>type(value)</b> $\rightarrow$ value's data type	
<b>len(seq)</b> - counts the number of top-level items of the sequence seq	
<b>abs, round</b> - Absolute value, nearest integer - ( <b>abs numerical_expression</b> ) - ( <b>round real_number_expression</b> )	
<b>max(seq), min(seq)</b> - maximum or minimum of the items in the sequence seq.	
<b>pow( base, exponent)</b> - base <sup>exponent</sup>	
<b>def</b> - function definition statement	
<b>lambda</b> - anonymous-function/closure definition expression - actually, it is the constructor of values of type function. <b>lambda param<sub>1</sub>, param<sub>2</sub>, ... , param<sub>N</sub> : body</b>	
<b>map</b> - <b>map(fun, arglist<sub>1</sub>, [(arglist<sub>2</sub>, ..., arglist<sub>n</sub>)])</b> - repeatedly applies a function fun on list(s) of items - a list of the results of the fun application is returned. <b>filter</b> - <b>filter(func, arglist)</b> - repeatedly applies a predicate func on each item of arglist - a list of the items that func holds, is returned. <b>eval</b> - <b>eval(e)</b> - evaluates the expression e that it receives as argument <b>reduce</b> - <b>reduce(func, arglist, [initval])</b> - repeatedly applies a function func on the list arglist, two items at-a-time, such that the result of each fun application substitutes those two items; the result is returned when the list is empty. It must be imported from the <b>functools</b> module.	
<b>expression1 == expression2</b> - equality comparison operator that returns True if the values of both expressions are equal <b>expression1 is expression2</b> - equality comparison operator that returns True if both expressions represent exactly the same data object <b>value in seq</b> - membership test operator that returns True if the value is found in the sequence seq.	
<b>map pattern expressed using List comprehension</b> - <b>[func(i) for i in iterable]</b> - a list is created as a result of applying the function func upon every item in the iterable.	
<b>filter pattern expressed using List comprehension</b> - <b>[i for i in iterable if funcp(i)]</b> - a list is created as a result of applying the boolean function funcp upon every item in the iterable; if funcp(i) holds, the item i is passed to the output list; otherwise, it is not passed to the output.	
<b>map-filter pattern expressed using List comprehension</b> - <b>[func(i) for i in iterable if funcp(i)]</b> - a list is created such that every item i in it is the value of func(i) if funcp(i) is True.	
<b>fobj = open(fname, 'r')</b> פתיחת קובץ לקריאה <b>fobj = open(fname, 'w')</b> פתיחת קובץ לכתיבה שימו לב שהפונקציה <b>open</b> מחזירה ערך מסוג קובץ ( <b>file</b> ) שהוא למעשה iterator שניתן להפעיל עליו לולאת <b>for</b> , בדיוק כמו על רשימה של מחזורות, שהן למעשה השורות שבקובץ.	
<b>lines = fobj.readlines()</b> קריאת כל הקובץ שורה אחת שורה לתוך המשתנה <b>lines</b> . כאשר <b>lines</b> יהיה רשימה של מחזורות, והמחזורות שמכילה את תוכן השורה ה- <b>i</b> בקובץ <b>fname</b> נמצאת בתוך <b>lines[i]</b>	
<b>fobj.close()</b> סגירת הקובץ	