

**CSI2520 - Projet intégrateur (Prolog/LP)**  
**Descriptions de mes prédicats**

NOTES IMPORTANTES :

1. Mon programme retourne les bonnes lettres et la bonne valeur mais knapsack\_2.txt est beaucoup trop grand et la récursivité en Prolog a du mal à s'en charger.
2. J'ai beaucoup utilisé [cette documentation \(SWI-Prolog\)](#).

Nom du prédicat	Description	Source
knapsack/7	Résout le problème de la même manière que celle faite en Programmation dynamique en Java mais avec le paradigme et la pensée de programmation logique	N/A
solveKnapsack/3	Fait lecture, résolution et écriture	N/A
getLetters/8	Trouve les lettres des items pris comme on doit faire dans la solution dynamique.	N/A
readKnapsackFile/6	Lis le fichier d'entrée, prépare les infos avec cleanKnapsackData/7 puis extract_1_from_1/6. Ici, extract_1_from_1 est utilisé pour extraire les noms, valeurs, weights d'items, ce qui est nécessaire pour procéder à la résolution du problème.	N/A
cleanKnapsackData/7	Prend la capacité et length des arrays et se sert de cleanItemsData/3 pour traiter les lignes avec les informations sur les items.	N/A
cleanItemsData/3	"Nétoie" les informations sur les items tel que(par exemple) "A 1 1" devient ["A", 1, 1].  <b>Voir commentaires pour plus de détails.</b>	N/A
file_lines/2	file_lines/2 lis un fichier donné et de met les informations qui y figurent dans une liste ligne par ligne dans l'ordre. Il se sert de stream_lines/2 pour faire ainsi.	<a href="#">Source</a>
stream_lines/2	stream_lines/2 sépare les lignes du fichier ligne par ligne et les met dans Lines.	<a href="#">Source</a>
writeToFile/3	Écris la solution valeur,lettres sur un fichier .sol du même nom	<a href="#">Source</a>
makeKTable/4	Constitue la table en s'aidant de makeNextRow/5. Le premier row est la rangée de 0 de longueur capacité étant la première valeur du paramètre AboveRow (AR).	N/A

makeNextRow/5	makeNextRow/5 constitue le prochain row de la KTable pour la solution dynamique a partir du dernier row fourni en paramètre.	<b>Source :</b> Inspiré d'un cours sur Zoom du professeur
getVW/6	getVW/6 prend les éléments tel que ValueIndex (VI) et WeightIndex (WI) sont égaux dans les listes qui contiennent ces informations.	N/A
getWWL/5	getWWL/5 constitue la liste des possibilités de paires [VALUE WEIGHT].	N/A
convert_ASCIIDEC_string/2	convert_ASCIIDEC_string/2 converti une liste de codes ASCII DECIMAUX en liste de string (ou inversement)	N/A
checkElements/4	Compare CurrentValue a la valeur juste au dessus dans la KTable BoolResult (B) = true si elles sont les mêmes.	N/A
getLastTwoRows/4	Donne les deux derniers rows a partir de l'index du dernier voulu dans KTable	N/A
equals/3	equals/3 unifie le 3ème param. a "t" si les deux premiers sont égaux, "f" sinon	N/A
max/3	Unifie le 3ème paramètre avec la plus grande valeur des 2 premiers paramètres.	N/A
fill/3	Donne une liste de longueur N, remplie de l'élément X	<a href="#">Source</a>
extract_1_from_1/6	Extrait chaque élément donc l'index est un multiple de D dans une liste L.	N/A
str_1_int_1/2	str_1_int_1/2 converti une liste de strings en une liste de ints (ou inversement)	N/A
sublist/4	Donne une sous liste d'une liste donnée	<a href="#">Source</a>