RAPPORT DE PROJET OCAML

MOUNSI AREZKI
FENNOUCH KOCEILA
DEVECIOGLU SERHAT

Répartition du travail :

MOUNSI Arezki : partie 1 et 2
FENNOUCH Koceila et Serhat DEVECIOGLU :
partie 3 et menu principale + aide partie 2

Documentation des fonctions :

* nombre : transforme type entier en type nombre

* nbr_of_lst : transforme une liste d'entier en liste de type nombre

* enlever : enlève un élément dans une liste

* combinaison : construit les combinaison possible deux à deux à partir d'une liste

* addition : fait l'addition de deux nombres et concatène le résultat a la liste passé en paramètre

* multiplication : fait la multiplication de deux nombres et concatène le resultat a la liste passé en paramètre

* soustraction : fait la soustraction de deux nombres et concatène le résultat à la liste passé en paramètre

* division : fait la division de deux nombres et concatène le résultat a la liste passé en paramètre

* appartient : vérifie si le nombre n'est pas déjà dans la liste

* fus : concatène les deux premiers élément d'une list list

* combin_abc : applique les opérations si dessous a 2 valeurs et les ajoute a la liste

* flat_map : c est l'équivalent de list.flatten et list.map combiner ensemble

* combin_ab_l : appliquer combin_abc a tous les éléments de la combinaison

* construire : construire les combinaisons possible à l'aide de tous les élément de la liste passer en paramètre

* liste_sol : liste les solutions

* to_string : affiche le type nombre comme string

* to_string_solution : affiche le type nombre en unit
* affich_combi_possibl : affiche tous les résultat de liste_sol
* best : renvoie la meilleur valeur proche d‘un nombre entre deux valeur
* meilleure_approximation : renvoie la plus proche valeur d‘un nombre parmi une liste de valeur
* trouve_sol : filtre une liste avec une valeur (=)
* foundp : renvoie true si il trouve un matching false sinon
* cher_val_aprox : elle applique un parcours tous en sauvegardant le plus proche nombre trouver
* la_meilleure_combinaison : rend la valeur la plus proche du nombre cible
* exception Not_foundd
* solutions_exact : elle renvoie l‘expression exact de la valeur cible false sinon
* list_of_string : transforme une chaîne de caractères en liste
* exception IllFormedExpression
* exception Depassement
* input_of_string : fonctions permetant de séléctionner un choix et appliquer l‘évaluation
* menu : lance le programme

Travail réalisé :

L'ensemble des 3 parties demandées ont été codées, ainsi que l'interface d'utilisation .

Mode d emploi :
menu :

```
######## MENU ########

#####     1 - Afficher toutes les combinaisons possibles          #####

#####     2 - la_meilleure_combinaison                            #####

#####     3 - resultat exact                                      #####

#####     4 - exit                                                #####
```

Entrer un entier entre 1 et 4
puis entrer l’entier cible puis la liste comme chaine de caractère :
Exemple:
Entrer un entier a évaluer : 17
entrer la liste de valeurs de départ : [10;7;2;5]
une valeur = 17 correspondant à l'expression " 10 + 7 " ‘

QUELQUE JEUX D'ESSAI :

étape 1 :

```
 1
Entrer une liste de depart de taille inferrieur a 4 svp [1;2;3]
une valeur = 6  correspondant a l'expression  " (1 + 2) + 3 "
une valeur = 9  correspondant a l'expression  " (1 + 2) * 3 "
une valeur = 1  correspondant a l'expression  " (1 + 2) / 3 "
une valeur = 4  correspondant a l'expression  " (2 - 1) + 3 "
une valeur = 2  correspondant a l'expression  " 3 - (2 - 1) "
une valeur = 6  correspondant a l'expression  " (1 + 3) + 2 "
une valeur = 8  correspondant a l'expression  " (1 + 3) * 2 "
une valeur = 2  correspondant a l'expression  " (1 + 3) - 2 "
une valeur = 4  correspondant a l'expression  " (3 - 1) + 2 "
une valeur = 1  correspondant a l'expression  " (3 - 1) / 2 "
une valeur = 6  correspondant a l'expression  " (2 + 3) + 1 "
une valeur = 4  correspondant a l'expression  " (2 + 3) - 1 "
une valeur = 7  correspondant a l'expression  " (2 * 3) + 1 "
une valeur = 5  correspondant a l'expression  " (2 * 3) - 1 "
une valeur = 2  correspondant a l'expression  " (3 - 2) + 1 "
```

étape2 :

```
 2
Entrer un entier a evaluer :  1999
entrer la liste de valeurs de depart :  [2;4;3;7;25]
une valeur = 2000  correspondant a l'expression  " (3 + 7) * ((2 * 4) * 25) "
```

étape3 :

```
 3
 Entrer un entier a evaluer :  782
entrer la liste de valeurs de depart :  [5;7;1;25;2]
une valeur = 782  correspondant a l'expression  " (25 - 2) * ((5 * 7) - 1) "
######## MENU ########
```