

Calculatrice en notation polonaise - Pour aller plus loin

Florian Bourse

1 Tester son programme

Pour tester son programme, on peut écrire deux fichiers contenant des exemples ainsi que le résultat attendu.

test.txt

```
42
+ 13 31
* 56 65
+ 41 * 2 3
+ + + 4 5 6 7
```

attentes.txt

```
42
44
3640
47
22
```

On peut ensuite lancer son programme avec pour entrée le fichier `test.txt` et écrire le résultat dans un fichier `result.txt` :

```
./calculatrice < test.txt > result.txt
```

puis comparer les résultats avec ceux attendus en utilisant la commande `diff` :

```
diff results.txt attentes.txt
```

Si les deux fichiers sont identiques, la commande ne renvoie rien, sinon on peut lire les erreurs :

résultat de la commande `diff`

```
2c2
< 70
---
> 44
4c4
< 70
---
> 47
```

Ici, la calculatrice a renvoyé 70 et 70 aux tests 2 et 4 à la place des résultats attendus 44 et 47.

2 Beyond `int`

Les entiers en OCaml étant limités à l'intervalle compris entre `min_int` et `max_int`, on a le comportement suivant :

nombre trop grand

```
# int_of_string "11111111111111111111111111111111";;  
Exception: Failure "int_of_string".
```

Cependant, il est tout à fait possible d'écrire une fonction

```
ajoute : string -> string -> string
```

qui nous donnerait :

```
# ajoute "11111111111111111111111111111111" "11111111111111111111111111111111";;  
- : string = "22222222222222222222222222222222"
```

On pourrait s'aider d'une fonction

```
ajoute_avec_retenue : string -> string -> char -> string
```

qui en plus des deux nombres garde en mémoire la retenue qui est 0 ou 1 pour l'addition.
De même pour la multiplication, la soustraction et la division.