Graf\_Calvin.cpp

#include <iostream>

***// J'ai tenté un truc mais pas terminé à cause du temps***

/\*

#define nombre(somme) \

#define afficher(s, somme) \

printf("Ligne %d : " nombre(somme) "\n", \_\_LINE\_\_, somme)

\*/

***// Je l'ai mis en %lld pour que tout fonctionne mais il faudrait modifier pour***

***// avoir soit %lld soit %d***

#define afficher(s, somme) \

printf("Ligne %d : %lld\n", \_\_LINE\_\_, somme)

***// Ne fonctionne pas car il faudrait trouver une solution pour***

***// implémenter ##typeMinMax. De cette manière, ça ne fonctionne pas.***

***// Cependant il s'agit peut-être de la meilleure solution.***

/\*

#define GENERIC\_SOMME(type, typeMinMax) \

type sommer(type n1, type n2) { \

if(n1 == ##typeMinMax || n2 == ##typeMinMax) { return ##typeMinMax; } \

else if(n1 == ##typeMinMax\_MAX || n2 == ##typeMinMax) { return ##typeMinMax; } \

int res = n1 + n2; \

return res; \

}

GENERIC\_SOMME(int, INT);

GENERIC\_SOMME(long long int, LONG\_LONG);

\*/

**long** **long** **int** sommer**(long** **long** **int** n1**,** **long** **long** **int** n2**)**

**{**

**if(**n1 **==** LONG\_LONG\_MIN **||** n2 **==** LONG\_LONG\_MIN**)** **{** **return** LONG\_LONG\_MIN**;** **}**

**else** **if(**n1 **==** LONG\_LONG\_MAX **||** n2 **==** LONG\_LONG\_MAX**)** **{** **return** LONG\_LONG\_MAX**;** **}**

**int** res **=** n1 **+** n2**;**

**return** res**;**

**}**

***// On duplique du code, probablement une meilleure solution pour éviter les***

***// fonctions ambiguës***

**long** **long** **int** sommer**(long** **long** **int** n1**,** **int** n2**)**

**{**

**if(**n1 **==** LONG\_LONG\_MIN **||** n2 **==** LONG\_LONG\_MIN**)** **{** **return** LONG\_LONG\_MIN**;** **}**

**else** **if(**n1 **==** LONG\_LONG\_MAX **||** n2 **==** LONG\_LONG\_MAX**)** **{** **return** LONG\_LONG\_MAX**;** **}**

**int** res **=** n1 **+** n2**;**

**return** res**;**

**}**

**int** sommer**(int** n1**,** **int** n2**)**

**{**

**if(**n1 **==** INT\_MIN **||** n2 **==** INT\_MIN**)** **{** **return** INT\_MIN**;** **}**

**else** **if(**n1 **==** INT\_MAX **||** n2 **==** INT\_MAX**)** **{** **return** INT\_MAX**;** **}**

**int** res **=** n1 **+** n2**;**

**return** res**;**

**}**

**int** main**(void)** **{**

afficher**(**"%d"**,** sommer**(**1**,** 2**));**

afficher**(**"%d"**,** sommer**(**INT\_MIN**,** **-**1**));**

afficher**(**"%d"**,** sommer**(**INT\_MAX**,** 1**));**

afficher**(**"%lld"**,** sommer**(**3LL**,** 4LL**));**

afficher**(**"%lld"**,** sommer**(**LLONG\_MIN**,** **-**1**));**

afficher**(**"%lld"**,** sommer**(**LLONG\_MAX**,** 1**));**

**const** **bool** INCLURE\_CAS\_SPECIAL **=** **true;**

***/\****

***if (INCLURE\_CAS\_SPECIAL)***

***afficher("%lld - %lld", sommer(INT\_MIN, -1LL), sommer(INT\_MAX, 1LL));***

***else***

***afficher("%s", "Calcul spécial pas inclus.");***

***\*/***

**return** EXIT\_SUCCESS**;**

**}**