

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;
}
```