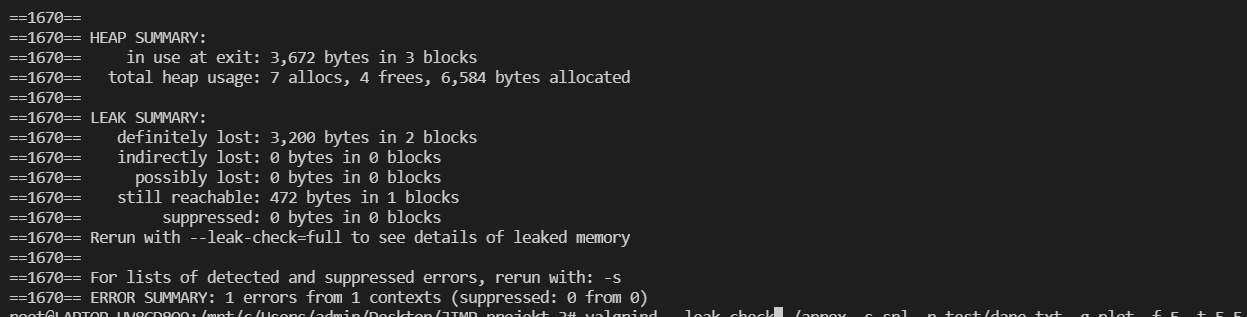
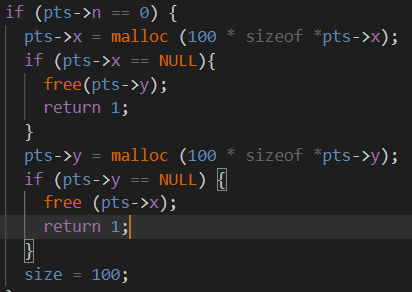
Podsumowanie wycieków pamięci:



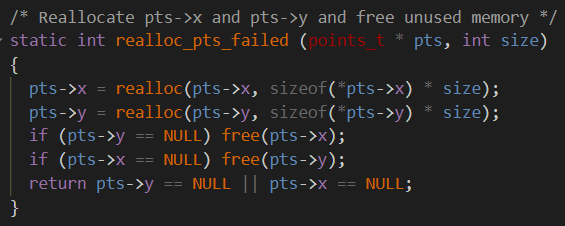
Dokładne miejsce wycieków:

Wyciek w points.c, 3200 bajtów wynika z tego, że po błędnej alokacji pamięci dla **pts->x**, lub **pts->y**, nie jest zwalniana pamięć dla tego “member”(członka?), dla którego pamięć została poprawnie zaalokowana. Jednocześnie utrata tych 3200 bajtów polega na błędnym działaniu funkcji **realloc\_pts\_failed**. W poprawionej wersji tej funkcji, skupiliśmy się na poprawnej realokacji pamięci do podanej na wejściu wielkości **size**, oraz na sprawdzeniu czy realokacja przebiegła pomyślnie poprzez porównanie wskaźników **pts->x**, oraz **pts->y** z wartością **NULL.**

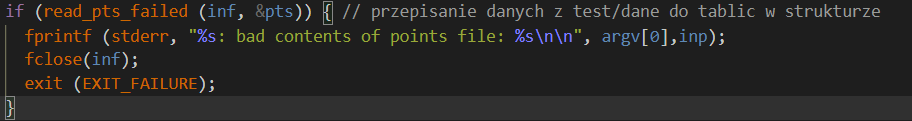
Poprawione działanie funkcji **read\_pts\_failed**:

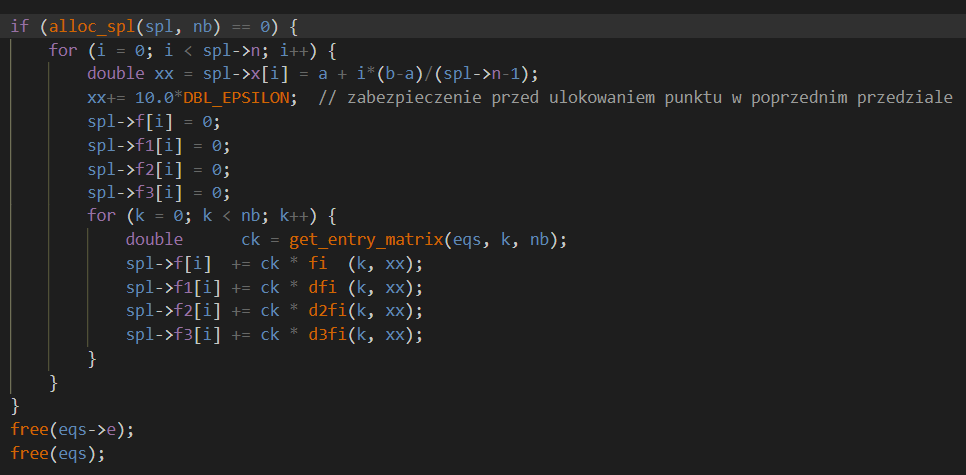


Poprawiona funkcja:



W main.c znajduje się wyciek polegający na nie zamknięciu otwartego wcześniej pliku, to powoduje że przy “wychodzeniu” z programu, cały czas mamy zaalokowaną pamięć na ten plik, czyli 472 bajty.



Przy poprawnym działaniu programu i poprawnych danych wejściowych, możliwa jest utrata danych przechowywanych w strukturze ‘**eqs**’, czyli macierzy przechowującej informacje dla folderu **/gaus** i nie tylko.

Dla ułatwienia obsługi pamięci struktury **spl**, zmieniliśmy funkcję **alloc\_spl,** w ten sposób aby ułatwić możliwość jej modyfikacji, oraz przejrzystość alokacji pamięci.

