We say that a store instruction such as "*p = v" is *silent* if *p already contains the same value that is in the variable v. The detection of silent stores, be it through the compiler, be it through the hardware, has already been the focus of much research. Silent stores are redundant, and can be bypassed in hardware, for instance.

```
int main() {
  int x[SIZE];
  for (int i = 0; i < SIZE; ++i) {
    x[i] = 0;
  }
  return 0;
}
```

```
int x[SIZE];
int main() {
  for (int i = 0; i < SIZE; ++i) {
    x[i] = 0;
  }
  return 0;
}
```

```
void inc(int* x, size_t SIZE) {
  for (int i = 0; i < SIZE; ++i) {
    ++x[i];
  }
}
```

1. Is there any store within the programs on the left that are VERY likely to be silent?
2. Is there any store within the programs on the left that are never silent?
3. Under which assumptions will the store in the `mul` function below be silent?
4. Can you actually conceive some stochastic analysis that predicts the likelihood of a store being silent?

```
void mul(
  int* x,
  int* y,
  int *z,
  size_t SIZE
) {
  for (int i = 0; i < SIZE; ++i)
    x[i] += y[i]*z[i];
}
```

The problem of estimating the chance that a store is silent is a nice example of how stochastic techniques play a role in the design of static program analysis!

Fernando Magno Quintão Pereira, Guilherme V. Leobas, Abdoulaye Gamatié: **Static Prediction of Silent Stores.** ACM Trans. Archit. Code Optim. 15(4): 44:1-44:26 (2019)

UFMG

DCC

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Compilers Laboratory