Lab 9 Assignment Report

Below, you can see the first and last versions of the changed codes as a result of research on Lab 9 and how to optimize over the internet.

1-

```
void opt_matMul(int **first, int **second, int **result, int size) {
  int i,j,k;
  // Initializing elements of matrix mult to 0.
  for (i = 0; i < size; ++i) {
    for (j = 0; j < size; ++j) {
      result[i][j] = 0;
    }
  }

  // Multiplying first and second matrices and storing it in result
  for (i = 0; i < size; i++) {
    for (j = 0; j < size; j++) {
      for (k = 0; k < size; k++) {
        result[i][j] += first[i][k] * second[k][j];
      }
    }
  }
}
```

Filling the array with 0 in the above function can be done later, as we will be searching the array below. I commented out the first for loop inside the function and filled it with 0 in the secondary for loop as you can see below.

```
void opt_matMul(int **first, int **second, int **result, int size) {
   int i,j,k;
   // Initializing elements of matrix mult to 0.
 /*  for (i = 0; i < size; ++i) {
      for (j = 0; j < size; ++j) {
         result[i][j] = 0;
      }
   }*/

   // Multiplying first and second matrices and storing it in result
   for (i = 0; i < size; i++) {
      for (j = 0; j < size; j++) {
              result[i][j] = 0;
         for (k = 0; k < size; k++) {
                 //result[i][j] = 0;

            result[i][j] += first[i][k] * second[k][j];
```

```
            }
        }
    }
}
```

2-

```
void opt_foo1(int **result, int size){

        int a = 1, b = 5, c = 25, d = 7, i, j;

        for (i = 0; i < size; i++)
                for(j = 0; j < size; j++)
                result[i][j] = (((c % d) * a / b) % d) * i;

}
```

In order to optimize the function, I saved time and space by doing the constant operation inside each for loop outside the for loop as you can see below.

```
void opt_foo1(int **result, int size){
        int a = 1, b = 5, c = 25, d = 7, i, j, e;
        e = (((c % d) * a / b) % d);
        for (i = 0; i < size; i++)
                for(j = 0; j < size; j++)
                result[i][j] = e * i;
}
```