| galois |

Exercise 1:

```
version 2.12.0
https://cryptol.net  :? for help

Loading module Cryptol
Cryptol> :l SHA256.cry
Loading module Cryptol
Loading module SHA256
SHA256> SHA256 "Hello World Folks"
0xd14155c5fb4dbbb2f8d1d3ade275982a610bc50ff85389a1093875b85993cfeb
SHA256>
```

Exercise 2:

```
[prompt]$ make sha256
cc sha256.c -o sha256
[prompt]$ sha256 "Hello World Folks" 17
d14155c5fb4dbbb2f8d1d3ade275982a610bc50ff85389a1093875b85993cfeb
[prompt]$
```

Exercise 3:

```
digest_in_bytes : {i} (fin i, 64 >= width (8*i)) => [i][8] -> [32][8]
digest_in_bytes msg = split(SHA256 msg)
```

```
version 2.12.0
https://cryptol.net  :? for help

Loading module Cryptol
Cryptol> :l SHA256.cry
Loading module Cryptol
Loading module SHA256
SHA256> digest_in_bytes "Hello World Folks"
[0xd1, 0x41, 0x55, 0xc5, 0xfb, 0x4d, 0xbb, 0xb2, 0xf8, 0xd1, 0xd3,
 0xad, 0xe2, 0x75, 0x98, 0x2a, 0x61, 0x0b, 0xc5, 0x0f, 0xf8, 0x53,
 0x89, 0xa1, 0x09, 0x38, 0x75, 0xb8, 0x59, 0x93, 0xcf, 0xeb]
SHA256>
```

Exercise 4:

```
SHA256_Buf_Wrapper(char *input, uint8_t digest[32]) {
   size_t len;
   for (len=0 ; ; len++) {
      if (input[len] == 0) break;
   }
   SHA256_Buf(input, len, digest);
}
```

```
[franco@franco lab5E]$ make sha256
cc sha256.c -o sha256
[franco@franco lab5E]$ sha256 "Hello World Folks"
d14155c5fb4dbbb2f8d1d3ade275982a610bc50ff85389a1093875b85993cfeb
[franco@franco lab5E]$
```

Exercise 5:

```
import "SHA256.cry";

let alloc_init ty v = do {
   p <- llvm_alloc ty;
   llvm_points_to p (llvm_term v);
   return p;
};

let ptr_to_fresh n ty = do {
   x <- llvm_fresh_var n ty;
   p <- alloc_init ty x;
   return (x, p);
};

let sha256_setup n = do {
   (m, pm) <- ptr_to_fresh "buffer" (llvm_array 32 (llvm_int 8));
   (input, pinput) <- ptr_to_fresh "input" (llvm_array n (llvm_int 8));
   llvm_execute_func [ pinput, pm ];
   llvm_points_to pm (llvm_term {{ digest_in_bytes pinput pm }});
};

let main = do {
   m <- llvm_load_module "sha256.bc";
   sha256_OK <- llvm_verify m "SHA256_Buf_Wrapper" [] false (sha256_setup 10) z3;
   print "Done!";
};
```