

```
-- Name: Xiaomeng
-- Email: xcao07@syr.edu
-- Section: CST 1-109
```

```
import Data.Char
```

```
duplicates :: String -> Bool
duplicates [] = False
duplicates (x:xs) = elem x xs || duplicates xs
```

```
zap :: Char -> String -> String
zap ch cs = [c | c <- cs, c /= ch]
```

```
unique :: String -> String
unique [] = []
unique (c:[]) = [c]
unique (c:cs)
    | elem c cs = unique (zap c cs)
    | otherwise = c : unique cs
```

```
prefix :: String -> String -> Bool
prefix xs [] = False
prefix [] ys = True
prefix (x:xs) (y:ys) = (x == y) && prefix xs ys
```

```
subseq :: String -> String -> Bool
subseq xs [] = False
subseq [] ys = True
subseq (x:xs) (y:ys) = (x == y && subseq xs ys) ||
(subseq (x:xs) ys)
```

```
substring :: String -> String -> Bool
substring (x:xs) [] = False
substring xs ys
    | prefix xs ys = True
    | substring xs (tail ys) = True
    | otherwise = False
```

```
subsequences :: String -> [String]
subsequences xs = [] : nonEmptySubsequences xs

nonEmptySubsequences :: String -> [String]
nonEmptySubsequences [] = []
nonEmptySubsequences (x:xs) = [x] : foldr f []
(nonEmptySubsequences xs)
  where f ys r = ys : (x : ys) : r
```