Assignment Set 3 (Functional Programming with Haskell)

Assignment – 1: Implement Haskell Functions for Basic Set Operations

A) Check whether a set is empty (NULL SET) --> checkNull
B) Union of two sets --> findUnion
C) Intersection of two sets --> findIntersection
D) Subtraction one set from another set --> findSubtraction
E) Addition of two sets. --> findAddition

```
*Main> findUnion [1,2,3] [4,5]
fromList [1,2,3,4,5]
*Main> findIntersection [1,2,3] [2,3]
fromList [2,3]
*Main> checkNull [2,3]
False
*Main> findSubtraction [1,2,3,4] [2,4]
fromList [1,3]
*Main> findAddition [1,2,3] [4,5,6]
fromList [5,6,7,8,9]
```

Assignment – 2: Billing System for Fast Food Stall

```
Enter code:
4719
Enter Quantity:
2
AVAILABLEMore Items? (Y/N):
Y
Enter code:
1234
Enter Quantity:
1
AVAILABLEMore Items? (Y/N):
N
ALCHERINGA 2018, STALL 14: TANGO FAST FOOD CENTER
4719     FISH FINGERS     121     2     242
1234     CHICKEN LOLLY     250     1     250
-------------------------------------------------
                                          492
Just "four hundred ninety-two"
```

Assignment – 3: Generating Cipher text from Plaintext and Vice Versa

3. A: Generating Cipher text from plain text

Algorithm:

- Take input plaintext string and the key
- check the first character of the plaintext , if it is uppercase letter, then go on the second character with same key.
- If charater is either a numerical (0-9) then follow the rules and map it to a character given in question.
- If none of the above cases, then find the addition of ASCII values of top character in the plaintext and key and wrap around if needed(when > 122).
- find the character corresponding to the added numer and it is map for that character.
- Do this recursively for all characters in plain text.

```
Enter the PlainText:
Hello90
Enter the key:
iitg
The Ciphertext is: Hfmxn&*
```

3. B: Generating Plaintext from Cipher text

Algorithm:

- Take input ciphertext string and the key
- check the first character of the plaintext , if it is uppercase letter, then go on the second character with same key.
- If charater is one of the mentioned in question, then follow the rules and map it to the number given in question.
- If none of the above cases, then find the subtraction of ASCII values of top character in the plaintext with key and wrap around if needed(when <97).
- find the character corresponding to the subtracted numer and it is map for that character.
- Do this recursively for all characters in cipher text.

```
Enter the Ciphertext:
Hfmxn&*
Enter the key:
iitg
The Plaintext is: Hello90
```