## **Ex. No. : 7.1** Binary String

**Program:** a = input() try:

c = int(a) print("Yes") except:

print("No")

## **Ex. No. : 7.2** DNA Sequence

**Program:** def findRepeatedSequences(s):

sequences = {} result = [] for i in range(len(s) - 9):

seq = s[i:i+10]

sequences[seq] = sequences.get(seq, 0) + 1 if sequences[seq] == 2: result.append(seq) return result s1 = input() for i in findRepeatedSequences(s1):

print(i)

## **Ex. No. : 7.3** American keyboard

**Program:**

def findWords(words): row1 = set('qwertyuiop') row2 = set('asdfghjkl') row3 = set('zxcvbnm')

result = [] for word in words:

w = set(word.lower()) if w.issubset(row1) or w.issubset(row2) or w.issubset(row3):

result.append(word) if len(result) == 0: print("No words") else:

for i in result: print(i)

a = int(input()) arr = [input() for i in range(a)] findWords(arr)

## **Ex. No. : 7.4** Print repeated no

**Program:**

n =input().split(" ") n = list(n) for i in range(len(n)):

for j in range(i+1,len(n)):

if n[i] == n[j]: print(n[i]) exit(0)

# **Ex. No. : 7.5** Check Pair

**Program:** def count\_distinct\_pairs(t, K):

distinct\_pairs = set() for i in range(len(t)):

for j in range(i + 1, len(t)):

if t[i] + t[j] == K:

distinct\_pairs.add((min(t[i], t[j]), max(t[i], t[j])))

return len(distinct\_pairs) t\_

input = input()

t = tuple(map(int, t\_input.split(',')))

K = int(input()) print(count\_distinct\_pairs(t, K))