

```

1  ✓ def demon(slayer):
2      tanjiro = lambda nezuko: nezuko(inosuke)
3      nezuko = lambda inosuke: tanjiro(slayer)
4      inosuke = 1
5      nezuko(6)
6
7  muzan = 6
8  yoriichi = 10
9  akaza = demon(lambda yoriichi: yoriichi - muzan)

```

Question #1

Global Frame:

Demon = func demon(slayer)
 Muzan = 6
 Yoriichi = 10
 Akaza = None

F1: demon

Slayer = lambda (yoriichi) line 9
 Nezuko = lambda (inosuke) line 3
 Inosuke = 1
 Tanjiro = lambda (yoriichi) line 9

F2: lambda (inosuke) line 3

Parent: F1

Inosuke = 6

F3: lambda (nezuko) line 2

Parent: F1

Nezuko = lambda (yoriichi) line 9

F4: lambda func line 9

Yoriichi = 1
 Return value = -5

Question #2

```
1  #Dragon Ball
2
3
4  def super(saiyan):
5      saiyan += 10000
6      return saiyan
7
8  def power_scanner(level):
9      if( level > 9000):
10         print("It's over 9000!")
11     else:
12         print("It's under 9000")
13     return level
14
15  def fight(player1, player2):
16      print("let the fight begin")
17      if player1 > player2:
18          print("Fighter 1 wins")
19      else:
20          print("Fighter 2 wins")
21
22  def power(up):
23      return lambda chi, kaioken: up(chi, kaioken)
24
25  def punched_by(hit):
26      return hit * .1
27
28  krillin = power(power(pow))
29  goku = 500
30  vegeta = 450
31  piccolo = 400
```

Expression and Output

power(vegeta)

Output:

Function

picolo = power(up)

fight(goku,picolo)

Output:

Error

goku = punched_by(vegeta)

Output:

455.0

krillin(100,2)

Output:

10000

Question #3

You are given a list of integers, "arr" and an integer 'k' and integer 'n'. Your goal is to find the first element kth subset where n is the number of subsets needed to be created.

```
1  def first_element(arr, k, n):  
2      subset_size = len(arr) // n  
3      subset_index = (k - 1) // subset_size  
4      element_index = (k - 1) % subset_size  
5      return arr[subset_index * subset_size + element_index]
```

Question #4

You are given a string "input" that contains a binary number. Your goal is to find the highest number of consecutive 1's within the string

```
1  ✓ def consecutive_ones(input):  
2      max_ones = 0  
3      count = 0  
4  ✓  for i in input:  
5  ✓      if i == '1':  
6          count += 1  
7  ✓      if count > max_ones:  
8          max_ones = count  
9  ✓      else:  
10         count = 0  
11     return max_ones
```