Quiz 10

Source code:

```
n = 5
W = 13
p = [20, 30, 35, 12, 3]
weight = [2, 5, 7, 3, 1]
p_by_w = [10, 6, 5, 4, 3]


class Priority_Queue:
    def __init__(self):
        self.pqueue = []
        self.length = 0

    def insert(self, node):
        for i in self.pqueue:
            get_bound(i)
        i = 0
        while i < len(self.pqueue):
            if self.pqueue[i].bound > node.bound:
                break
            i+=1
        self.pqueue.insert(i,node)
        self.length += 1

    def print_pqueue(self):
        for i in list(range(len(self.pqueue))):
            print ("pqueue",i, "=", self.pqueue[i].bound)

    def remove(self):
        try:
            result = self.pqueue.pop()
            self.length -= 1
        except:
            print("Queue is empty, cannot remove from empty list.")
        else:
            return result

class Node:
    def __init__(self, level, profit, weight):
        self.level = level
        self.profit = profit
        self.weight = weight
        self.items = []
```

```
def get_bound(node):
    if node.weight >= W:
        return 0
    else:
        result = node.profit
        j = node.level + 1
        totweight = node.weight
        while j <= n-1 and totweight + weight[j] <= W:
            totweight = totweight + weight[j]
            result = result + p[j]
            j+=1
        k = j
        if k<=n-1:
            result = result + (W - totweight) * p_by_w[k]
        return result


nodes_generated = 0
pq = Priority_Queue()

v = Node(-1, 0, 0)
nodes_generated+=1
maxprofit = 0
v.bound = get_bound(v)


pq.insert(v)

while pq.length != 0:

    v = pq.remove()


    if v.bound > maxprofit:
        u = Node(0, 0, 0)
        nodes_generated+=1
        u.level = v.level + 1
        u.profit = v.profit + p[u.level]
        u.weight = v.weight + weight[u.level]
        u.items = v.items.copy()
        u.items.append(u.level)


        if u.weight <= W and u.profit > maxprofit:
            maxprofit = u.profit
            bestitems = u.items
```

```
        u.bound = get_bound(u)
        if u.bound > maxprofit:
            pq.insert(u)

        u2 = Node(u.level, v.profit, v.weight)
        nodes_generated+=1
        u2.bound = get_bound(u2)
        u2.items = v.items.copy()

        if u2.bound > maxprofit:
            pq.insert(u2)



print("\nMaxprofit = ", maxprofit, "\nSelected nodes = ", nodes_generated)
print("Items selected = ", bestitems)
```

Output:

harshavaidhyam@Harshas-MacBook-Pro quiz 10 % cd /Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ Design/quiz\ 10 ; /usr/bin/env /usr/local/bin/python3 /Users/harshavaidhyam/.vscode/extensions/ms-python.python-2022.16.1/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 62874 -- /Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ Design/quiz\ 10/q10.py

Maxprofit = 70
Nodes generated = 13
Bestitems = [0, 2, 3, 4]