

Assignment-8

Dynamic Programming

Qn1:

Code:

```
n = int(input())
wt = list(map(int,input().split()))
val = list(map(int,input().split()))
w = int(input())
dp = [[0 for i in range(w+1)] for j in range(n+1)]
def ks(w,wt,val,n):
    if n==0 or w==0:
        return 0
    if dp[n][w]:
        return dp[n][w]
    if wt[n-1]<=w:
        dp[n][w] = max(val[n-1]+ks(w-wt[n-1],wt,val,n-1),ks(w,wt,val,n-1))
    else:
        dp[n][w] = ks(w,wt,val,n-1)
    return dp[n][w]
print('max value:',ks(w,wt,val,n))
for i in dp:
    print(*i)

def ele():
    global w
    res = dp[n][w]
    W = w
    picked = []
    for i in range(n, 0, -1):
        if res <= 0:
            break
        if res == dp[i - 1][W]:
            continue
```

```

        else:
            picked.append(wt[i - 1])
            res = res - val[i - 1]
            W = W - wt[i - 1]

    print(*picked)
ele()

```

Qn2:

Code:

```

V = int(input())
INF = 99999

def floydWarshall(graph):
    dist = list(map(lambda i: list(map(lambda j: j, i)), graph))
    for k in range(V):
        for i in range(V):
            for j in range(V):
                dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])
    printSolution(dist)

def printSolution(dist):
    for i in dist:
        print(i)

graph = []
print("enter 9999 if no edge is present")
for i in range(V):
    arr = map(int, input().split())
    graph.append(arr)
floydWarshall(graph)

```