```
# Structure for an item which stores weight and

# corresponding value of Item

class Item:

        def __init__(self, value, weight):

                self.value = value

                self.weight = weight



def knapSack_01(W, arr):

    n=len(arr)

    K = [[0 for x in range(W + 1)] for x in range(n + 1)]



    # Build table K[][] in bottom up manner

    for i in range(n + 1):
```

```
    for w in range(W + 1):

        if i == 0 or w == 0:

            K[i][w] = 0

        # elif wt[i-1] <= w:

        elif arr[i-1].weight <= w:

            wt=arr[i-1].weight

            val=arr[i-1].value

            K[i][w] = max(val + K[i-1][w-wt],  K[i-1][w])

        else:

            K[i][w] = K[i-1][w]



return K[n][W]
```

```
# Driver's Code

if __name__ == "__main__":

    W = 8

    arr = [Item(1, 2), Item(2, 3), Item(5, 4), Item(6, 5)]

    print("Knapsack Capacity:", W)

    print("Maximum Profit:", knapSack_01(W, arr))




# val = [3, 4, 5, 6]

# wt = [2, 3, 4, 5]
```