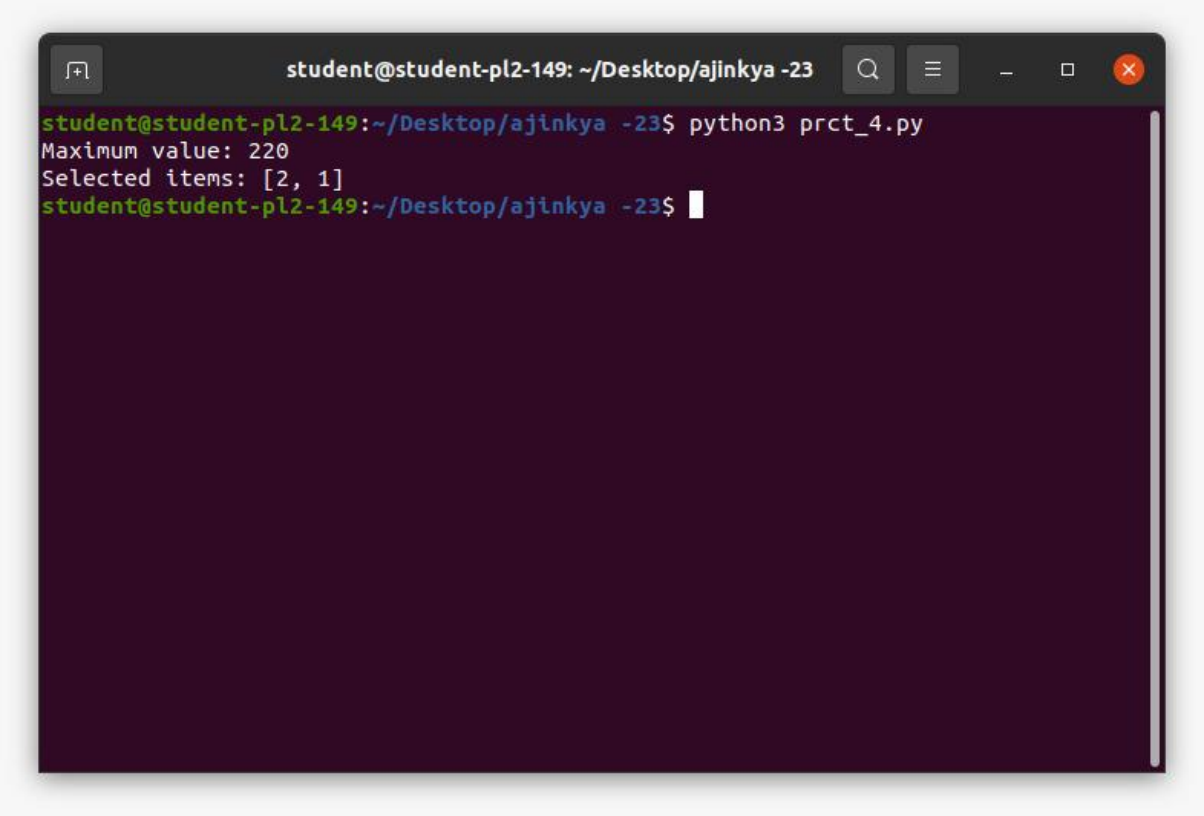
Practical No. 4

Code :

```
def knapsack_01(n, values, weights, W):
dp = [[0] * (W+1) for _ in range(n+1)]

for i in range(n+1):
for w in range(W+1):
if i == 0 or w == 0:
dp[i][w] = 0
elif weights[i-1] <= w:
dp[i][w] = max(dp[i-1][w], dp[i-1][w-weights[i-1]] + values[i-1])
else:
dp[i][w] = dp[i-1][w]

selected_items = []
i, w = n, W
while i > 0 and w > 0:
if dp[i][w] != dp[i-1][w]:
selected_items.append(i-1)
w -= weights[i-1]
i -= 1

return dp[n][W], selected_items

if __name__ == "__main__":
n = 3
values = [60, 100, 120]
weights = [10, 20, 30]
W = 50

max_value, selected_items = knapsack_01(n, values, weights, W)
print("Maximum value:", max_value)
print("Selected items:", selected_items)
```

Output :


student@student-pl2-149: ~/Desktop/ajinkya -23
student@student-pl2-149:~/Desktop/ajinkya -23$ python3 prct_4.py
Maximum value: 220
Selected items: [2, 1]
student@student-pl2-149:~/Desktop/ajinkya -23$