Source Code:

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
int k[100][100];
int knapsack(int n, int capacity, int weight[], int value[]){
    if(n==0 || capacity==0)return 0;
    if(k[n-1][capacity] != -1)return k[n-1][capacity];
    if(weight[n - 1] <= capacity){
        int op1 = knapsack(n-1, capacity - weight[n-1], weight, value) + value[n-1];
        int op2 = knapsack(n-1, capacity, weight, value);
        return k[n-1][capacity] = max(op1,op2);
    }
    else {
        int op1 = knapsack(n-1, capacity, weight, value);
        return k[n-1][capacity] = op1;
    }
}
int main(){

    int n; cin >> n;
    int weight[n], value[n];
    for (int i = 0; i < n; i++) {
        cin >> weight[i];
    }
    for (int i = 0; i < n; i++) {
        cin >> value[i];
    }
    int capacity; cin >> capacity;
```

```
for(int i = 0; i <= n; i++){
    for(int j = 0; j <= capacity; j++){
        if(i==0||j==0) k[i][j] = 0;
    }
}
for(int i = 1; i <= n; i++){
    for(int j = 1; j <= capacity; j++){
        if(weight[i-1] <= j) k[i][j] = max(k[i-1] [j-weight[i-1]] + value[i-1], k[i-1][j]);
        else k[i][j] = k[i-1][j];
    }
}

int i = n, j = capacity;
while(i!=0 and j!=0 ){
    if(k[i][j] != 0){
        if(k[i][j]==k[i-1][j]){
            i--;
            continue;
        }
        else if(k[i][j]==k[i][j-1]){
            j--;
            continue;
        }
        else{
            cout << weight[i-1] << " ";
            i--;
            j--;
        }
    }
```

```
    }
    cout << endl << k[n][capacity] << endl;
return 0;
}
```

Input - Output: