

```
/*knapsack 0/1 (global decleration)*/
```

```
#include <stdio.h>
```

```
int knapsack_0_1(int w[], int p[]);
```

```
int n = 0, c = 0; // Global declaration
```

```
int main() {
```

```
    int i;
```

```
    int w[20], p[20];
```

```
    printf("Enter the array size for both weight & profit(no. of objs & profits:");
```

```
    scanf("%d", &n);
```

```
    printf("\nEnter the weights and profits");
```

```
    for (i = 0; i < n; i++) {
```

```
        printf("\nWeight[%d]:", i + 1);
```

```
        scanf("%d", &w[i]);
```

```
        printf("Profit[%d]:", i + 1);
```

```
        scanf("%d", &p[i]);
```

```
    }
```

```
    printf("\nEnter the capacity:");
```

```
    scanf("%d", &c);
```

```
    int max = knapsack_0_1(w, p);
```

```
    printf("Max profit: %d\n", max);
```

```
    return 0;
```

```
}
```

```
int knapsack_0_1(int w[], int p[]) {
```

```
    int i, j;
```

```
    int ks[n + 1][c + 1];
```

```

for (i = 0; i <= c; i++) {
    ks[0][i] = 0;
}

for (i = 0; i <= n; i++) {
    ks[i][0] = 0;
}

for (i = 1; i <= n; i++) {
    for (j = 1; j <= c; j++) {
        if ((w[i - 1] <= j) && ((p[i - 1] + ks[i - 1][j] - w[i - 1]) > ks[i - 1][j])) {
            ks[i][j] = p[i - 1] + ks[i - 1][j] - w[i - 1];
        } else {
            ks[i][j] = ks[i - 1][j];
        }
    }
}

printf("The matrix is\n");
for (i = 0; i <= n; i++) {
    for (j = 0; j <= c; j++) {
        printf("%d ", ks[i][j]);
    }
    printf("\n");
}

// Backtrack to find the selected items
i = n;
j = c;
while (i > 0 && j > 0) {
    if (ks[i][j] != ks[i - 1][j]) {

```

```

        printf("Object %d selected (Weight: %d, Profit: %d)\n", i, w[i - 1], p[i - 1]);

        j -= w[i - 1];

        i--;
    } else {

        i--;
    }
}

return ks[n][c];
}

```

The screenshot shows a Windows command prompt window with the following text:

```

C:\Users\HP\OneDrive\Desktop >
Enter the array size for both weight & profit(no. of objs & profits):3

Enter the weights and profits
Weight[1]:1
Profit[1]:10

Weight[2]:2
Profit[2]:12

Weight[3]:4
Profit[3]:28

Enter the capacity:6
The matrix is
0 0 0 0 0 0
0 10 10 10 10 10
0 10 12 22 22 22
0 10 12 22 28 38 40
Object 3 selected (Weight: 4, Profit: 28)
Object 2 selected (Weight: 2, Profit: 12)
Max profit: 40

-----
Process exited after 20.87 seconds with return value 0
Press any key to continue . . .

```

The taskbar at the bottom shows the system clock as 13:27 on 06-12-2023, with weather information (74°F Cloudy) and various application icons.