KNAPSACK ALGORITHM:

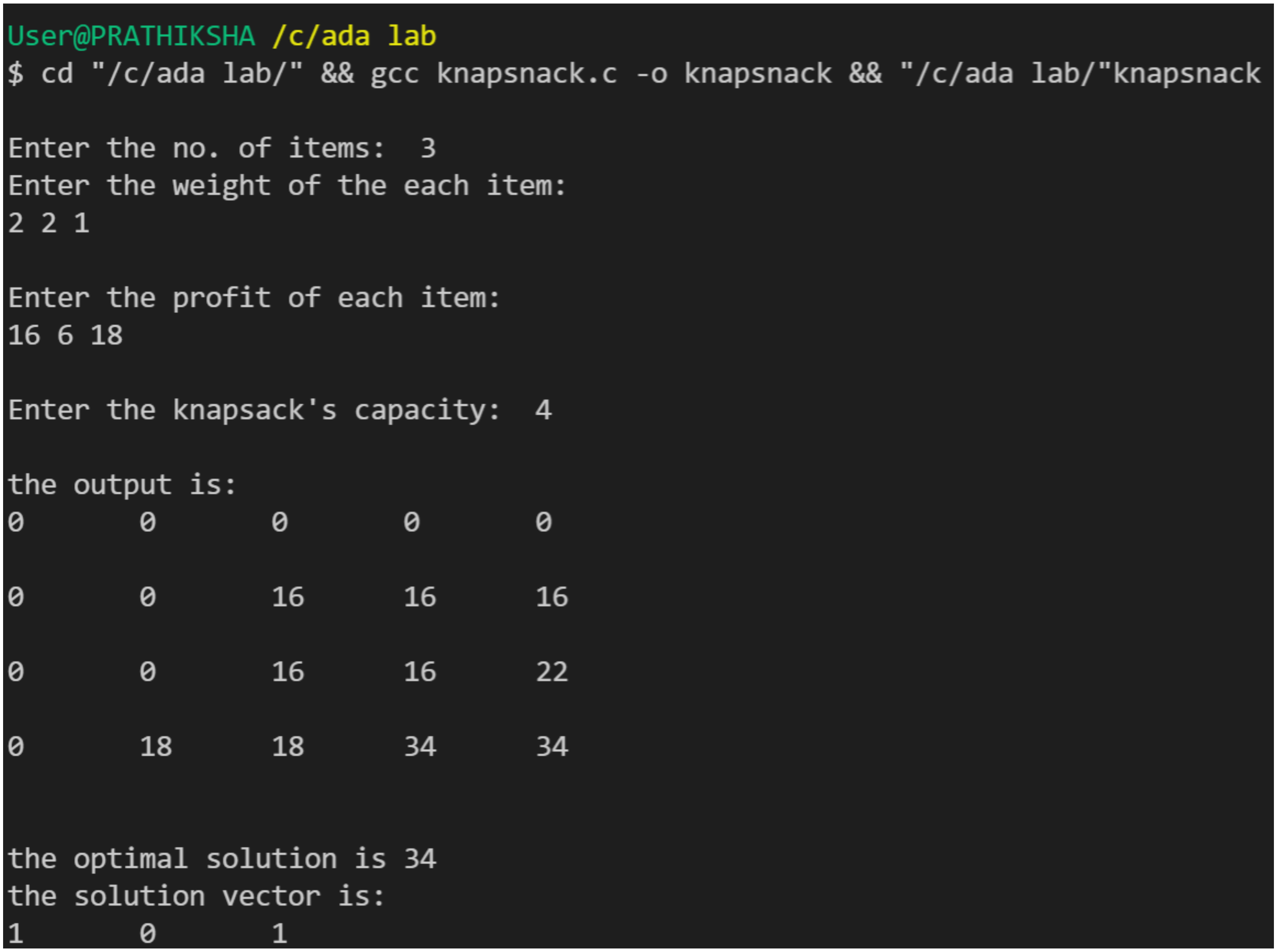
PROGRAM:

```
#include <stdio.h>
#include <conio.h>
void knapsack();
int max(int, int);
int i, j, n, m, p[10], w[10], v[10][10];
void main()
{
    printf("\nEnter the no. of items:  ");
    scanf("%d", &n);
    printf("Enter the weight of the each item:\n");
    for (i = 1; i <= n; i++)
    {
        scanf("%d", &w[i]);
    }
    printf("\nEnter the profit of each item:\n");
    for (i = 1; i <= n; i++)
    {
        scanf("%d", &p[i]);
    }
    printf("\nEnter the knapsack's capacity:\t");
    scanf("%d", &m);
    knapsack();
}

void knapsack()
{
    int x[10];
    for (i = 0; i <= n; i++)
    {
        for (j = 0; j <= m; j++)
        {
            if (i == 0 || j == 0)
            {
                v[i][j] = 0;
            }
            else if (j - w[i] < 0)
            {
                v[i][j] = v[i - 1][j];
            }
            else
            {
                v[i][j] = max(v[i - 1][j], v[i - 1][j - w[i]] + p[i]);
            }
        }
    }
    printf("\nthe output is:\n");
    for (i = 0; i <= n; i++)
    {
        for (j = 0; j <= m; j++)
        {
            printf("%d\t", v[i][j]);
        }
        printf("\n\n");
    }
    printf("\nthe optimal solution is %d", v[n][m]);
    printf("\nthe solution vector is:\n");
    for (i = n; i >= 1; i--)
    {
        if (v[i][m] != v[i - 1][m])
        {
            x[i] = 1;
            m = m - w[i];
        }
        else
        {
            x[i] = 0;
        }
    }
    for (i = 1; i <= n; i++)
    {
        printf("%d\t", x[i]);
    }
}

int max(int x, int y)
{
    if (x > y)
    {
        return x;
    }
    else
    {
        return y;
    }
}
```

OUTPUT:



```
User@PRATHIKSHA /c/ada lab
$ cd "/c/ada lab/" && gcc knapsnack.c -o knapsnack && "/c/ada lab/"knapsnack

Enter the no. of items:  3
Enter the weight of the each item:
2 2 1

Enter the profit of each item:
16 6 18

Enter the knapsack's capacity:  4

the output is:
0       0       0       0       0

0       0       16      16      16

0       0       16      16      22

0       18      18      34      34


the optimal solution is 34
the solution vector is:
1       0       1
```