```cpp
/*Knapsack using dynamic approach*/

#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    int n;
    int m;

    cout << "Enter the value of n : ";
    cin >> n;

    int profit[n];
    int weight[n];
    int x[n];

    cout << "\nEnter the weight a bag can hold : ";
    cin >> m;

    int v[n+1][m+1];

    for(int i=0;i<n+1;i++)
    {
        for(int w=0;w<m+1;w++)
        {
            v[i][w]=0;
        }
    }

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

    cout << "Enter profits : ";
    for(int i=0;i<n;i++)
    {
        cin >> profit[i];
    }

    cout << "Enter weights : ";
    for(int i=0;i<n;i++)
    {
        cin >> weight[i];
    }

    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n-i-1;j++)
        {
            if(weight[j]>weight[j+1])
            {

                int temp1=weight[j];
                weight[j]=weight[j+1];
                weight[j+1]=temp1;

                int temp2=profit[j];
                profit[j]=profit[j+1];
                profit[j+1]=temp2;
            }
        }
    }

    for(int i=0;i<n+1;i++)
    {
        for(int w=0;w<m+1;w++)
        {
            if(i==0 || w==0)
```

```
            {
                v[i][w]=0;
            }
            else if(weight[i-1]<=w)
            {
                v[i][w]=max(v[i-1][w],((v[i-1][w-weight[i-1]])+profit[i-1]));
            }
            else
            {
                v[i][w]=v[i-1][w];
            }
        }
    }

    for(int i=0;i<n+1;i++)
    {
        for(int w=0;w<m+1;w++)
        {
           cout << v[i][w] << "\t";
        }
        cout << endl;
    }

    int p=v[n][m];
    int flag=0;
    int q=n-1;

    for(int i=n;i>=0;i--)
    {
        for(int j=0;j<m+1;j++)
        {
           if(v[i-1][j] == p)
           {
               flag=0;
               break;
           }
           else
           {
               flag=1;
           }
        }

        if(flag==0)
        {
           x[q]=0;
           q--;
        }
        else
        {
           x[q]=1;
           p=p-profit[q];
           q--;
        }
    }

    for(int i=0;i<n;i++)
    {
        cout << x[i] << "\t";
    }

    int pr=0;
    for(int i=0;i<n;i++)
    {
        pr=pr+(x[i]*profit[i]);
    }

    cout << endl << "Profit : " << pr << endl;

    return 0;
}
```