

Write a function `int Knapsack(int W, int[] weights, int[] values)` in C# that determines the maximum value of items that can fit into a knapsack with a capacity `W`. The function should handle up to 100 items. Find the optimal way to fill the knapsack with the given items to achieve the maximum total value. You must consider that you cannot break items, but have to include them whole.

```
public class KnapsackDemo {

    static int knapSack(int M,int[] w,int[] p,int n) {

        int[][] C=new int[n+1][M+1];

        for(int i=1;i<=n;i++) {

            for(int j=1;j<=M;j++) {

                if(w[i-1] > j)

                    C[i][j]=C[i-1][j];

                else {

                    C[i][j]=Math.max(p[i-1]+C[i-1][j-w[i-1]],

                                    C[i-1][j]);

                }

            }

        }

        return C[n][M];

    }

    public static void main(String[] args) {

        int[] p= {60,100,120};

        int[] w= {10,20,30};

        int M=50;

        int n=p.length;

        System.out.println("The maximum profit : "+knapSack(M,w,p,n));

    }

}
```

