

## 一、编程题

### ACM：高效货运

题目描述：1.老李是货运公司承运人，老李的货车额定载货重量为wt  
2.现有两种货物，货物A单件重量为wa，单件运费利润为pa，货物B单件重量为wb，单件运费利润为pb  
3.老李每次发车时载货总重量刚好为货车额定载货重量wt，车上必须同时有货物A和货物B，货物A、B不可切割  
4.老李单车次满载运输可获得最高利润是多少

输入描述：第一列输入为货物A的单件重量wa， $0 < wa < 10000$   
第二列输入为货物B的单件重量wb， $0 < wb < 10000$   
第三列输入为货车的额定载重wt， $0 < wt < 100000$   
第四列输入为货物A的单件运费利润pa， $0 < pa < 1000$   
第五列输入为货物B的单件运费利润pb， $0 < pb < 1000$

输出描述：单次满载运输的最高利润

补充说明：

示例1  
输入：10 8 36 15 7  
输出：44  
说明：

示例2  
输入：1 1 2 1 1  
输出：2  
说明：

代码：

```
import java.util.*;
```

// 注意类名必须为 Main，不要有任何 package xxx 信息

```
public class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int a = sc.nextInt();  
        int b = sc.nextInt();  
        int c = sc.nextInt();  
        int d = sc.nextInt();  
        int e = sc.nextInt();  
        int m = c / a;  
        int n = 0;  
        if (c - a * m >= b) {  
            n = (c - a * m) / b;  
        }  
        Map<Integer, Integer> map = getItem(m, n, a, b, c);  
        int result = 0;  
        Set<Integer> set = map.keySet();  
        for (Integer i : set) {
```

```

        int temp = i * d + map.get(i) * e;
        result = Math.max(result, temp);
    }
    System.out.println(result);
}

public static Map<Integer, Integer> getItem(int m, int n, int a, int b, int c) {
    Map<Integer, Integer> map = new HashMap<>();
    if (n != 0) {
        map.put(m, n);
    }
    for (int i = m - 1; i > 0; i--) {
        if (c - i * a >= b) {
            n = (c - i * a) / b;
        }
        if (n != 0) {
            map.put(i, n);
        }
    }
    return map;
}
}

```