题目描述：

在一款虚拟游戏中生活，你必须进行投资以增强在虚拟游戏中的资产以免被淘汰出局。现有一家 Bank，它提供有若干理财产品 m，风险及投资回报不同，你有 N（元）进行投资，能接受的总风险值为 X。

你要在可接受范围内选择最优的投资方式获得最大回报。

说明：

在虚拟游戏中，每项投资风险值相加为总风险值；

在虚拟游戏中，最多只能投资 2 个理财产品；

在虚拟游戏中，最小单位为整数，不能拆分为小数；

投资额*回报率=投资回报

输入描述：

第一行：产品数(取值范围[1, 20])，总投资额(整数，取值范围[1, 10000])，可接受的总风险(整数，取值范围[1, 200])

第二行：产品投资回报率序列，输入为整数，取值范围[1,60]

第三行：产品风险值序列，输入为整数，取值范围[1,100]

第四行：最大投资额度序列，输入为整数，取值范围[1,10000]

输出描述：

每个产品的投资额序列

补充说明：

在虚拟游戏中，每项投资风险值相加为总风险值；

在虚拟游戏中，最多只能投资 2 个理财产品；

在虚拟游戏中，最小单位为整数，不能拆分为小数；

投资额*回报率=投资回报

示例 1

输入：

```
5 100 10
10 20 30 40 50
3 4 5 6 10
20 30 20 40 30
```

输出：

```
0 30 0 40 0
```

说明：

投资第二项 30 个单位，第四项 40 个单位，总的投资风险为两项相加为 4+6=10

```
import java.util.Arrays;
import java.util.Scanner;

/**
 * @author zyd
 * @Create 2023/11/7-19:57
 * @Description
 */
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        // 注意 hasNext 和 hasNextLine 的区别
        // 注意 while 处理多个 case
        String[] s1 = in.nextLine().split(" ");
        int num = Integer.parseInt(s1[0]);
        int money = Integer.parseInt(s1[1]);
        int risk = Integer.parseInt(s1[2]);

        String[] str1 = in.nextLine().split(" ");
        String[] str2 = in.nextLine().split(" ");
        String[] str3 = in.nextLine().split(" ");

        int[] huibaolv = new int[num];
        int[] fengxian = new int[num];
        int[] touzie = new int[num];
        int[] res = new int[num];
        int[] res2 = new int[num];
        int total = 0;
        int max = 0;

        for (int i = 0; i < num; i++) {
            huibaolv[i] = Integer.parseInt(str1[i]);
```

```
            fengxian[i] = Integer.parseInt(str2[i]);
            touzie[i] = Integer.parseInt(str3[i]);
            res[i] = 0;
        }

//          System.out.println(num);
//          System.out.println(money);
//          System.out.println(risk);
//          System.out.println(Arrays.toString(huibaolv));
//          System.out.println(Arrays.toString(fengxian));
//          System.out.println(Arrays.toString(touzie));
        for (int i = 0; i < num; i++) {
            int sum = 0;
            if (fengxian[i] <= risk && touzie[i] <= money) {
                sum += touzie[i] * huibaolv[i];
            } else if (fengxian[i] <= risk && touzie[i] > money){
                sum += money * huibaolv[i];
            }
            if (sum > max) {
                max = sum;
                Arrays.fill(res,0);
                res[i] = touzie[i];
            }
        }
        for (int i = 0; i < num; i++) {
            for (int j = i+1; j < num; j++) {
                int sum = 0;
                if (fengxian[i] + fengxian[j] <= risk && touzie[i] + touzie[j] <= money) {
                    sum += touzie[i] * huibaolv[i];
                    sum += touzie[j] * huibaolv[j];
                    if (sum > max) {
                        max = sum;
                        Arrays.fill(res,0);
                        res[i] = touzie[i];
                        res[j] = touzie[j];
                    }
                } else if (fengxian[i] + fengxian[j] <= risk && touzie[i] + touzie[j] > money){
                    int diff = touzie[i] + touzie[j] - money;
                    int value = 0;
                    for (int k = 0;k <= diff;k++){
                        int sum2 = 0;
                        if (touzie[i] - k >=0 && touzie[j] - (diff -k) >=0){
                            sum2 += (touzie[i] - k) * huibaolv[i];
                            sum2 += (touzie[j] - (diff -k)) * huibaolv[j];
```

```
                                }
                                if (sum2 > sum) {
                                        sum = sum2;
                                        value    = k;
                                }
                        }
                        if (sum > max) {
                                max = sum;
                                Arrays.fill(res,0);
                                res[i] = touzie[i] - value;
                                res[j] = touzie[j] - (diff -value);
                        }
                }
        }
    }

        for (int i = 0; i < num;i++){
                System.out.print(res[i]+" ");
        }
    }
}
```