```java
 1: import java.io.BufferedReader;
 2: import java.io.InputStreamReader;
 3: import java.math.BigInteger;
 4: import java.util.Arrays;
 5:
 6:
 7: public class Lunch {
 8:
 9:         static int[] arra;
10:         static int[] arrb;
11:         static int[] arrc;
12:         static int[] arrd;
13:
14:         static long total;
15:
16:         public static void main(String[] args) {
17:
18:                 BufferedReader bfr = new BufferedReader(new InputStreamReader(System.in));
19:
20:                 try {
21:
22:                         String line = bfr.readLine();
23:                         while (line != null) {
24:
25:                                 if (line.startsWith("0 0 0 0 0")) break;
26:
27:                                 int l, a,b,c,d;
28:                                 l = Integer.parseInt(line.split(" ")[0]);
29:
30:                                 a = Integer.parseInt(line.split(" ")[1]);
31:                                 b = Integer.parseInt(line.split(" ")[2]);
32:                                 c = Integer.parseInt(line.split(" ")[3]);
33:                                 d = Integer.parseInt(line.split(" ")[4]);
34:
35:                                 arra = new int[a];
36:                                 arrb = new int[b];
37:                                 arrc = new int[c];
38:                                 arrd = new int[d];
39:
40:                                 String[] split = bfr.readLine().split(" ");
41:                                 for (int i=0; i<a; i++) {
42:                                         arra[i] = Integer.parseInt(split[i]);
43:                                 }
44:                                 split = bfr.readLine().split(" ");
45:                                 for (int i=0; i<b; i++) {
46:                                         arrb[i] = Integer.parseInt(split[i]);
47:                                 }
48:                                 split = bfr.readLine().split(" ");
```

```java
49:                        for (int i=0; i<c; i++) {
50:                                arrc[i] = Integer.parseInt(split[i]);
51:                        }
52:                        split = bfr.readLine().split(" ");
53:                        for (int i=0; i<d; i++) {
54:                                arrd[i] = Integer.parseInt(split[i]);
55:                        }
56:                        /*
57:                        Arrays.sort(arra);
58:                        Arrays.sort(arrb);
59:                        Arrays.sort(arrc);
60:                        Arrays.sort(arrd);
61:                        */
62:                        /*
63:                        System.out.println(Arrays.toString(arra));
64:                        System.out.println(Arrays.toString(arrb));
65:                        System.out.println(Arrays.toString(arrc));
66:                        System.out.println(Arrays.toString(arrd));
67:                        */
68:                        total = 0;
69:
70:                        check(l+1);
71:
72:
73:                        //System.out.println(total);
74:
75:                        bfr.readLine();
76:                        line = bfr.readLine();
77:                    }
78:
79:                } catch (Exception e) {
80:                        e.printStackTrace();
81:                }
82:
83:        }
84:
85:        private static void check(int limit) {
86:
87:                int[][] arr = new int[limit+1][4];
88:                long total = 0;
89:                BigInteger t = new BigInteger("0");
90:
91:                for (int i=0; i<arra.length; i++) {
92:                        arr[arra[i]][0]++;
93:                }
94:
95:                for (int i=0; i<arrb.length; i++) {
96:                        for (int j=0; j<limit; j++) {
```

```java
 97:                              long count = arr[j][0];
 98:                              if (count == 0) continue;
 99:                              int nprice = j + arrb[i];
100:                              if (nprice > limit) continue;
101:
102:                              arr[nprice][1] += count;
103:                      }
104:
105:                  }
106:
107:              for (int i=0; i<arrc.length; i++) {
108:                      for (int j=0; j<limit; j++) {
109:                              long count = arr[j][1];
110:                              if (count == 0) continue;
111:                              int nprice = j + arrc[i];
112:                              if (nprice > limit) continue;
113:
114:                              arr[nprice][2] += count;
115:                      }
116:
117:                  }
118:
119:              for (int i=0; i<arrd.length; i++) {
120:                      for (int j=0; j<limit; j++) {
121:                              long count = arr[j][2];
122:                              if (count == 0) continue;
123:                              int nprice = j + arrd[i];
124:                              if (nprice > limit) continue;
125:
126:                              arr[nprice][3] += count;
127:                      }
128:
129:                  }
130:
131:              for (int i=0; i<limit; i++) {
132:                      t = t.add(new BigInteger(arr[i][3] + ""));
133:                  }
134: /*
135:              for (int j=0; j<4; j++) {
136:                      for (int i=0; i<limit; i++) {
137:                              System.out.print(arr[i][j] + " ");
138:                      }
139:                      System.out.println();
140:              }
141:              */
142:              System.out.println(t.toString());
143:
144:          }
```

```
145:
146: }
147:
148:
149:
150: /*
151:
152: 11 3 1 1 1
153: 4 5 6
154: 3
155: 2
156: 1
157:
158: 10 4 5 4 2
159: 3 2 5 7
160: 1 1 8 4 2
161: 3 5 2 1
162: 2 3
163:
164: 10 4 5 4 4
165: 3 2 5 7
166: 1 1 8 4 2
167: 3 5 2 1
168: 2 3 0 0
169:
170: 0 0 0 0 0
171:
172:  */
```