IOI'98 – Setubal/Portugal

Problem Portfolio

Starry Night

1. Introduction to the Problem

Starry Night is a graphic pattern matching problem, algorithmically challenging, but allowing a natural gradation on the degree of completion of the solution.

The classification given by the Portuguese Scientific Committee was as follows:

Problem type:	Geometric
Problem understanding	Easy
Algorithm effort	Medium
Implementation effort	Medium
Number of possible solutions per test	1
Several levels of testing	Yes

The task description is easy to understand. Outlining a solution is also not too difficult.

However, due to the multiple orientations of the clusters and their irregular shapes, the algorithmic details of any solution get very complex. So, considering the time constrains, it is hard to develop a complete solution for this problem.

2. Algorithms

2.1 Overall description

Any full solution requires the development of the following two main algorithms.

- Cluster detection: This amounts to the classical problem of detecting the connected components of a graph;

- Cluster comparison: Two clusters A and B are compared according to the 8 possible combinations of rotation/reflection

2.2 Cluster comparison

The best way to perform the comparison of cluster “A” with cluster “B” is associate, successively, 8 different co-ordinate systems with the cluster “A”. Clusters may depict irregular shapes. Hence, it is also necessary to compute the 8 starting stars of the “A” cluster where the comparison with “B” cluster must start for each of the 8 orientations. A brute force approach would try *every* star in cluster “A” as *candidate* starting point for the comparison with cluster “B”.

There are alternative methods for comparing cluster “A” with cluster “B”. For example, it is possible to copy the cluster “B” to a separated buffer and change successively its orientation in order to compare it with cluster “A”. This method is slower and algorithmically more laborious.

The program does not need to keep a record for every cluster in the map. In fact, while scanning the map, it is enough to record one *representative* cluster for each distinct shape that comes across. The clusters that are not *representative* are simply *marked* with the low case letter corresponding to their shape.

2.3 Implementation

The example of an implementation is given in the PASCAL program named "starry.p" and in C program named "starry.c".

3. Testing

3.1 Output filter

In this problem, the input does not determine the output uniquely. Because of this the following "filter" was developed. It normalises the output allowing it to be directly compared with the official solution:

```
// StNorm.c    by Artur Miguel Dias/1998

// Normalise a sky map

#include <stdio.h>

main() {
      char dic[128], name = 'a' ;
      int i, c ;

      for( i = 0 ; i < 128 ; i++ )
             dic[i] = '\0' ;
      dic[' '] = ' ' ;
      dic['\n'] = '\n' ;

      while( (c = getchar()) != EOF ){
             if( dic[c] == '\0' )
                    dic[c] = name++ ;
             putchar(dic[c]) ;
      }
}
```

3.2 Test grades

The challenge is to advance as far as possible in the completion of a solution. The first four input files are used to rank the solutions according to the criteria:

File1: one cluster only;
File2: multiple clusters, all distinct;
File3: multiple clusters, some repetitions, but all the repetitions with the same orientation;
File4: multiple clusters, some repetitions, various orientations.

The input file 5 presents a large and complex map and attempts to establish the total correctness of a solution. It includes: several error prone patterns, a number of clusters (401) near the allowed maximum, the maximum number of distinct shapes, one cluster with the maximum number of stars.

File5: as in File4 but very large and complex map.

Partial solutions should be within the reach of most of the competitors, though.

Files 1 and 2: fairly easy, even if some ingenuity is required (difficulty: 2 out of 5);
File 3: mildly difficult: (difficulty: 3 out of 5);

Files 4 and 5: in practice, very hard due to time constraints (difficulty: 4.5 out of 5)