

I. Lý thuyết: (4.0 đ)

Phát biểu đúng mỗi câu → 1 điểm

II. Bài toán: (16.0 đ)

1. Điểm yên ngựa (5.0 đ)

❖ Thuật toán:

- Nếu M_i là Min của các phần tử A_{ij} ở dòng i , còn M_j là Max của các phần a_{ij} ở cột j thì $M_i \leq A_{ij} < M_j$.
- Điều này cho phép ta có một vài nhận xét sau đây :
 - a/ Min ở bất kỳ dòng nào không lớn hơn Max ở bất kỳ cột khác, nghĩa là luôn có: $M_i < M_j$.
 - b/ Nếu đối với bất kỳ i và j nào thoả mãn đẳng thức $M_i = M_j$ thì Max của Min trùng với Min của Max, còn của giao của dòng i với cột j sẽ có điểm yên ngựa.
 - $M_i = A[i,j] = M_j$ (*)
 - c/ Nếu điểm $A[i,j]$ là điểm yên ngựa (*) được thoả mãn và suy ra Max của Min bằng Min của Max. Việc còn lại là thực thi các điểm nói trên trong chương trình. ở mỗi dòng chúng ta tìm Min M_i , trong số các Min này chọn ra Max → Ma và ghi lại dòng chứa nó. Việc tìm Min ở dòng này sẽ dừng lại nếu nó nhỏ hơn giá trị Ma hiện có.
 - Sau đó tìm dòng có chứa phần tử Max bằng giá trị tìm được Ma . Nếu cột như thế không có thì không có điểm yên ngựa. Thực tế ở chương trình chúng ta sử dụng điều là nếu ở cột không có phần tử lớn hơn Ma thì cột đó sẽ là cột phải tìm theo lý do nêu trong mục a/.
- **(Có nhiều cách giải khác)**

2. Làng trung tâm

Dạng đề mở, đánh giá mức độ tư duy bậc cao: Tư duy phê phán, khả năng loại bỏ, ra quyết định chọn lựa, giải quyết vấn đề nhằm tìm ra giải pháp cho một vấn đề cụ thể trong cuộc sống.

3. Bài toán sau mô tả một thuật toán mã hoá đơn giản (6.0 điểm)

Program BaiTap;

Uses sysUtils;

```

Var k: integer;
    plainText,
    cipherText,
    testText: String;
Procedure ReadFile(nameFile: String; Var k: integer; Var plainText: String);
Var inFile: Text;
Begin
    Assign(inFile, nameFile);
    Reset(inFile);
    Readln(inFile, k);
    Readln(inFile, plainText);
    Close(inFile);
End;

Procedure WriteFile(nameFile: String; txt: String);
Var outFile: Text;
Begin
    Assign(outFile, nameFile);
    Rewrite(outFile);
    Writeln(outFile, txt);
    Close(outFile);
End;

Function Encode(k: integer; plainText: String): String;
Var cText: String;
    i: integer;
Begin
    cText := "";
    plainText := UpperCase(plainText);
    for i:= 1 to length(plainText) do
        Begin
            cText := cText + Chr((((Ord(plainText[i]) - Ord('A') + k) mod 26) + Ord('A')));
        End;
    Encode := cText;
End;

Function Decode(k: integer; cipherText: String): String;

```

```

Var pText: String;
    i: integer;
Begin
    pText := "";
    for i:= 1 to length(cipherText) do
        Begin
            pText := pText + Chr((((Ord(cipherText[i]) - Ord('A') + 26 - k) mod 26) +
Ord('A')));
        End;
        Decode := pText;
    End;
End;

```

```

Begin
    Writeln('CHUONG TRINH MA HOA TEXT');
    ReadFile('mahoa.inp', k, plainText);
    cipherText := Encode(k, plainText);
    WriteFile('encode.out', cipherText);
    testText := Decode(k, cipherText);
    WriteFile('decode.out', testText);
    Writeln('Finished, Enter to Exit..');
    Readln();

```

End.