Universidade Federal de Viçosa
Centro de Ciências Exatas
Departamento de Informática

2ª PROVA DE PROGRAMAÇÃO I – INF 110 (30 pontos) – 2015

NOME_____MAT_____

1. (5 pontos) Um palíndromo é uma sequência de caracteres que pode ser lida da esquerda para direita ou da direita para esquerda. Por exemplo, *"Amor, Roma"* é um palíndromo. O programa abaixo deveria verificar se uma dada palavra é um palíndromo. Nesse programa o usuário fornece como entrada o número de caracteres que a palavra contém e, em seguida, digita a palavra. O programa, no entanto, contém erros de programação.

   Descreva no espaço abaixo os erros contidos no programa. Podem-se assumir apenas entradas com caracteres minúsculos. E mais, não é preciso reescrever o código, mas apenas identificar os erros (através da numeração das linhas) e explicar como corrigi-los.

```
01 #include <iostream>
02 #include <cstring>
03
04 using namespace std;
05
06 int main() {
07    int n;
08    cin >> n;
09    char * palavra = new char[n+1];
10    cin >> palavra;
11    bool pal = false;
12    for(int i = 0, j = n; i<j; i++, j--) {
13        if(palavra[i] != palavra[j]) {
14            pal = true;
15            break;
16        }
17    }
18    if(pal) cout << "PALÍNDROMO" << endl;
19    else cout << "NÃO PALÍNDROMO" << endl;
20    return 0;
21 }
```

Descreva aqui o erros do programa acima e explique como corrigi-los:

2. (6 pontos) Escreva uma função em C/C++ que recebe como parâmetro uma matriz `M` de inteiros, inteiros `n` e `m` indicando o número de linhas e colunas de `M`, respectivamente, e um valor `x` a ser buscado em `M`. A função deve retornar os valores `i` e `j` indicando, respectivamente, a linha e a coluna de `x` na matriz. Caso `x` não esteja na matriz, então retorne os valores `-1` e `-1` para linha e coluna. Haverá no máximo uma ocorrência de `x` em `M`. A matriz `M` é recebida como parâmetro através de uma variável do tipo `int**`.

   O retorno dos valores de `i` e `j` pode ser implementado através de variáveis passadas por referência para a função, ou através de um tipo heterogêneo de dados (THD). Caso escolha utilizar um THD, escreva o trecho de código definindo o THD. Não é necessário implementar a função `int main()`.

Escreva aqui a sua solução:

3. (6 pontos) A função `int _strcmp(const char* s1, const char *s2)` recebe como parâmetro duas strings, `s1` e `s2`, e retorna um inteiro `m` indicando o seguinte:

   `m = -1`, se `s1` é menor que `s2`

   `m = 1`, se `s1` é maior que `s2`

   `m = 0`, se `s1` e `s2` são iguais.

   Onde as relações menor, maior e igual obedecem a ordem lexicográfica dos caracteres. Implemente a função `_strcmp` em C/C++.

Escreva aqui a sua solução:

4. (6 pontos) O programa abaixo recebe como entrada `n` pares de notas e matrículas de alunos da UFV. Esses dados são armazenados em variáveis do tipo `Aluno`, que contém os campos `mat` (matrícula) e `nota` (nota). O programa armazena esses `n` pares em um arranjo que é então ordenado de acordo com o critério a ser implementado em uma função de comparação.

   Escreva a função de comparação de forma que ela funcione com o código abaixo (fique atento com relação à assinatura da função). O código abaixo (juntamente com a sua função de comparação) deve imprimir os pares matrícula e nota ordenados por nota (decrescente) e, em caso de empate, por matrícula (crescente).

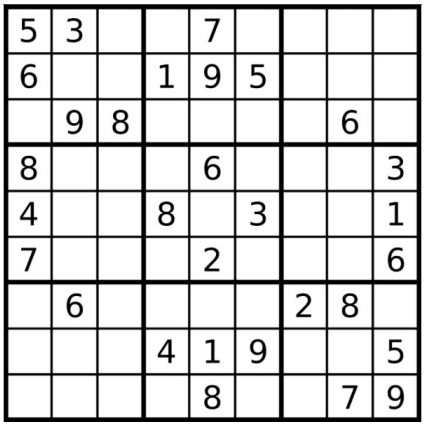
```
#include<iostream>

using namespace std;

struct Aluno {
    int mat;
    int nota;
};

void ordenar(Aluno alunos[], int n, bool (*compare)(Aluno, Aluno)) {
    for(int i = n - 1; i > 0; i--) {
        for(int j = 0; j < i; j++) {
            if(compare(alunos[j], alunos[j+1])) {
                Aluno aux = alunos[j];
                alunos[j] = alunos[j+1];
                alunos[j+1] = aux;
            }
        }
    }
}

int main() {
    Aluno alunos[100];
    int n;
    cin >> n;
    for(int i = 0; i < n; i++)
        cin >> alunos[i].mat >> alunos[i].nota;
    ordenar(alunos, n, compare_nota);
    for(int i = 0; i < n; i++)
        cout << alunos[i].mat << " " << alunos[i].nota << endl;
}
```

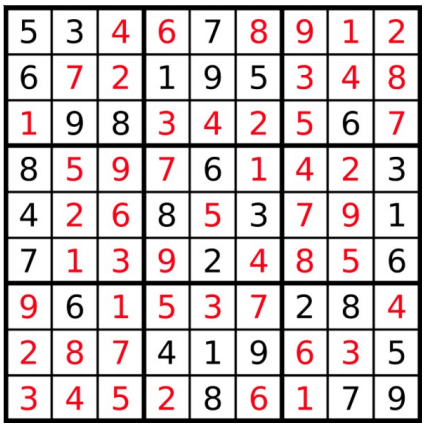
Escreva aqui a sua solução:

5. (7 pontos) O Sudoku é um quebra cabeça muito popular – jornais mundialmente famosos publicam com regularidade instâncias do quebra cabeça. A figura abaixo à esquerda mostra um exemplo de uma instância de Sudoku. O problema é representado por uma matriz 9x9 com algumas das posições preenchidas com inteiros de 1 a 9. O objetivo é preencher todas as posições da matriz (também com inteiros de 1 a 9). O preenchimento da matriz deve obedecer às seguintes restrições:

   a) Não pode haver números repetidos em nenhuma linha da matriz.
   b) Não pode haver números repetidos em nenhuma coluna da matriz.
   c) Não pode haver números repetidos nas sub-matrizes 3x3 separadas pelas linhas mais grossas da matriz do quebra cabeça.

   A imagem à direita mostra uma solução para o problema mostrado na imagem à esquerda. Repare que as três restrições acima são satisfeitas na solução do quebra cabeça.





Fonte das imagens: https://en.wikipedia.org/wiki/Sudoku

Escreva um programa em C/C++ que recebe como entrada uma solução do Sudoku (matriz 9x9 preenchida) e verifica se a solução fornecida como entrada é válida. O seu programa deve imprimir na tela a mensagem `SIM` caso a solução fornecida como entrada satisfaça as três regras descritas acima; o programa deve imprimir a mensagem `NÃO` caso contrário.

Escreva a sua solução no verso dessa folha.