

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
 Algoritmos e Estruturas de Dados I – 1º. Semestre 2010 – 2a. Prova

1. Um quadrado latino pode ser representado por um arranjo de arranjos de int `ql2d[N][N]`, também pode ser representado por um “cubo de incidência” int `ql3d[N][N][N]` onde `ql3d[i][j][k]=TRUE` se e somente se `ql2d[i][j]=k+1`. Complete o trecho abaixo que converte um quadrado latino em um cubo de incidência.

```
for(i=0; i<N; i++)
  for(j=0; j<N; j++)
    for(k=0; k<N; k++)
      ql3d[i][j][k]=FALSE;
for(i=0; i<N; i++)
  for(j=0; j<N; j++) _____;
```

2. Considere os dois trechos de programa abaixo que verificam se uma certa representação de sudoku em um arranjo 9x9 tem valores distintos nas linhas. Explique em termos da questão de generalidade da representação dos valores porque o trecho (i) deve ser preferido.

(i)	(ii)
<pre>for(i=0; i<9; i++){ for(j=0; j<9; j++){ v=s[i][j]; for(k=j+1; k<9; k++) if(v==s[i][k]){ printf("Repete o digito!\n"); return FALSE; } } }</pre>	<pre>for(i=0; i<9; i++){ for(k=1; k<10; k++) digVisto[k]=FALSE; for(j=0; j<9; j++){ if(digVisto[s[i][j]]){ printf("Repete o digito!\n"); return FALSE; } digVisto[s[i][j]]=TRUE; } }</pre>

3. Complete o trecho da função que irá retornar TRUE no caso de um sudoku válido e FALSE caso contrário.

```
for(i=0; i<9; i++){
  for(j=1; j<10; j++) digVisto[j]=FALSE;
  baseLin=3*(i/3); baseCol=3*(i%3);
  for(m=baseLin; m<baseLin+3; m++)
    for(n=baseCol; n<baseCol+3; n++){
      if(_____(a)_____){
        printf("repete digito");
        return FALSE;
      }
      digVisto[_____(b)_____] = TRUE;
```

4. Complete o trecho de programa abaixo que calcula uma resposta do jogo Mastermind (Clássico) como sendo um valor cuja dezena corresponde aos pinos pretos e a unidade corresponde aos pinos brancos.

```
int ed[4]={1,1,1,1};
int td[4]={1,1,1,1};
int uresp=0, dresp=0;
for(i=0; i<4; i++)
  if(e[i]==t[i]) dresp++;
for(i=0; i<4; i++)
  for(j=0; j<4; j++)
    if(_____) {
      uresp++;
      ed[i]=FALSE;
      .....td[j]=FALSE;
    }
return dresp*10 + (uresp-dresp);
```

5. Complete o trecho de programa abaixo que copia a “entrada padrão” na “saída padrão”.

```
int c
c=getchar();
while(c!=EOF){
    ____ (a) ____;
    ____ (b) ____;
}
```

6. Explique porque a classe abaixo não compila.

```
class A{private: int x; public: static void f(int x){this->x=x;}};
```

7. (a)O que é escrito em função da execução da chamada da função f(x)?

```
class A {public: virtual void m(){cout<<"A\n";}};
class B :public A{public: void m(){A::m(); cout<<"B\n";}};
class C :public B{public: void m(){B::m(); cout<<"C\n";}};
void f(A &a){a.m();}
...
C x; f(x);
```

- (b)Complete o trecho de programa em C++ abaixo onde a classe Base não pode ter instâncias e a classe Derivada deve poder ter instâncias (A solução não deve utilizar restrição de acesso ao construtor).

```
class Base{_____.}
class Derivada :public Base{_____.
```

8. Faça um desenho, seguindo as convenções de aula, da estrutura criada a partir da variável pb.

```
class Nodo(int info;
public:
    Nodo *prox;
    static int cnt;
    Nodo(int i, Nodo *p){info=i; prox=p; cnt++;}};
int Nodo::cnt=0;
...
Nodo a(10,NULL);
Nodo *pb=new Nodo(20, &a);
```