

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
 Algoritmos e Estruturas de Dados – 1º. Semestre 2012 – 1a. Prova

1. Mostre o que é impresso na execução do trecho de programa abaixo (observe os espaços!):

```
int a[]={4,0,2,3,1}; int N=5;
int ii,jj;
for(ii=0; ii<N; ii++) printf("%d ", a[ii]);
printf("\n");
for(ii=0; ii<N; ii++){
  for(jj=0; jj<N; jj++){
    if (a[jj]==ii&&ii!=jj) printf("* ");
    else printf(". ");
  }
  printf("\n");
}
```

2. Mostre 1 (um) exemplo do que é escrito na execução da chamada `tstRand(90)`.

```
void tstRand(int n){ //considere rand() entre 0 e 32767
  double atual=rand()/32767., prox=rand()/32767.;
  int t1t1=0, t1t2=0, t1t3=0, t2t1=0, t2t2=0, t2t3=0, t3t1=0, t3t2=0, t3t3=0;
  int ii;
  for(ii=0; ii<n; ii++){
    if(atual<=1./3.)
      if(prox<=1./3.) t1t1++; else if(prox<=2./3.)t1t2++; else t1t3++;
    if(atual>1./3.&atual<=2./3.)
      if(prox<=1./3.) t2t1++; else if(prox<=2./3.)t2t2++; else t2t3++;
    if(atual>2./3)
      if(prox<=1./3.) t3t1++; else if(prox<=2./3.)t3t2++; else t3t3++;
    atual=prox;
    prox=rand()/32767.;
  }
  printf("%d %d %d %d %d %d %d %d %d\n",
    t1t1, t1t2, t1t3, t2t1, t2t2, t2t3, t3t1, t3t2, t3t3 );
}
```

3. Mostre as expressões (a) e (b) de indexação do arranjo `t` que torna a função abaixo equivalente a função acima:

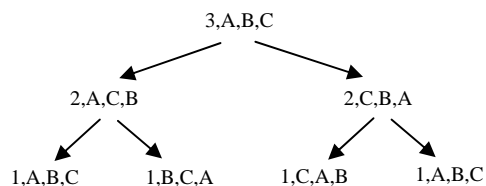
```
void tstRandM(int n){
  double atual=rand()/32767., prox=rand()/32767.;
  int t[3][3]={0,0,0,0,0,0,0,0,0};
  int ii, jj;
  for(ii=0; ii<n; ii++){
    t[atual<=1./3?0:_____ (a) _____][_____ (b) _____]++;
    atual=prox;
    prox=rand()/32767.;
  }
  for(ii=0; ii<3; ii++) for(jj=0; jj<3; jj++) printf("%d ", t[ii][jj]);
  printf("\n");
}
```

4. Faça um desenho, conforme a convenção de sala de aula, que mostra as estruturas em memória construídas a partir apenas das sentenças abaixo. A questão será avaliada em função do desenho mostrar de forma clara o que é `a1` e `a2`.

(i) `char *a1[4];` (iii) `char (*a2)[4]`

5. Considere a função recursiva apresentada abaixo. Mostre o que é escrito na execução da chamada

```
h(3, 'A', 'B', 'C')
void h(int n, char f, char d, char a){
  if(n==1)
    printf("%c -> %c\n", f, d);
  else{
    h(n-1, f, a, d);
    printf("%c -> %c\n", f, d);
    h(n-1, a, d, f);
  }
}
```



6. Considere que $a[i][j]=v$ modela que v alunos escolheram a opção j da questão i de uma prova. Complete o trecho de programa abaixo que preenche o arranjo gf . O elemento $gf[k]=x$ indica que a letra 'A'+ x é a opção mais frequente para a questão k .

```
//a[i][j]= numero de alunos que responderam letra 'A'+j na questão i
static int gf[5];
int indMaior, i, j;
for(i=0; i<MAXQUEST; i++){ //iteracao de questoes
    indMaior=0;
    for(j=0; j<MAXOP; j++) // iteracao de opcoes
        if(_____(a)_____) _____(b)_____;
    gf[i]=indMaior;
}
```

7. Um quadrado latino pode ser representado por um arranjo de arranjos de $int\ q12d[N][N]$, também pode ser representado por um “cubo de incidência” $int\ q13d[N][N][N]$ onde $q13d[i][j][k]=TRUE$ se e somente se $q12d[i][j]=k+1$. 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++)
            q13d[i][j][k]=FALSE;
for(i=0; i<N; i++)
    for(j=0; j<N; j++) _____;
```

8. Considere um sudoku “clássico” (conforme material da disciplina) codificado no arranjo $int\ s[9][9]$; Complete o trecho abaixo da função que faz várias verificações e retorna FALSE caso encontre algum tipo de invalidade.

```
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;
        }
}
```