

```

1a float m[3][3];
1 ocupa 3x3x4=36 bytes na memoria
b
2 21 = 16+4+1
  = 0x128 + 0x64 + 0x32 + 1x16 + 0x8 + 1x4 + 0x2 + 1x1
  = 00010101
  122 = 64+32+16+8+2
  = 0x128 + 1x64 + 1x32 + 1x16 + 1x8 + 0x4 + 1x2 + 0x1
  = 01111010
  a b | a OR b
  -----
  0 0 | 0
  0 1 | 1
  1 0 | 1
  1 1 | 1
  21 | 122 = 01111111
  = 0x128 + 1x64 + 1x32 + 1x16 + 1x8 + 1x4 + 1x2 + 1x1
  = 127
3a typedef struct {
    char marca[20];
    char modelo[20];
    int ano;
    char matricula[9];
    int ultinspec;
} carro;
3 carro carros[1000];
b
3c void mostraano(int anox, carro carros[1000]) {
    int i;
    for (i=0; i<1000; i++)
        if (carros[i].ano == anox)
            printf("%s %s %s\n", carros[i].matricula,
                carros[i].marca, carros[i]. modelo);
    }
4a float frecuencia(float lambda) {
    return( 3.0e8/lambda );
}
4 void gama(float lamda) {
b    if (lambda>0.01)
        printf("Radio Frecuencias");
    else
        if (lambda>1.0e-3)
            printf("Micro Ondas");
        else
            if (lambda>1.0e-6)
                printf("Infra Vermelhos");
            else
                if (lambda>300.0e-9)
                    printf("Luz Visivel");
                else
                    if (lambda>10.0e-9)

```

```

        printf("Ultra Violetas");
    else
        if (lambda>10.0e-12)
            printf("Raios X");
        else
            if (lambda>10.0e-15)
                printf("Raios Gama");
            else
                printf("Raios Cosmicos");
    }
5a int mult(int a, int b) {
    int i, m=0;
    for (i=1; i<=a; i++)
        m += b; // ou: m = m+b;
    return(m);
}
5 int mult(int a, int b) {
b     if (a==0)
        return(0)
    else
        return(b + mult(a-1, b));
}
6 vantagens: rapidez, flexibilidade
desvantagens: mais complicado (mais facil escrever erros)

```