

ALUNO(A): _____ MATRICULA: _____ NOTA: _____

1) [15 pontos] Considere o seguinte programa:

```

1 #include <stdio.h>
2 # define N 3
3 # define M 3
4 void funcaoA(int v[][N]);
5 void funcaoB(int s[][N],int t)
6 {
7     int i,j;
8     funcaoA(s);
9     for(i=1; i<t; i++)
10         for(j=0; j<N; j++)
11             s[i][j] = s[i-1][j] * 2;
12 }

```

```

12 void funcaoA(int v[][N])
13 {
14     int i;
15     for(i=0; i<N; i++)
16         scanf("%d",&v[0][i]);
17 }
18 int main()
19 {
20     int d[M][N];
21     funcaoB(d,2);
22     printf("%d\n", d[0][2]);
23     printf("%d\n", d[1][1]);
24     printf("%d\n", d[1][0]);
25     return 0;
26 }

```

Assumindo que as entradas fornecidas para o programa foram os valores 5 6 7, indique na tabela abaixo:

- ocupação da matriz do programa principal ao final da execução (use ? para valores não conhecidos);
- saída do programa, no formato exato da impressão;
- sequência que indica a ordem de execução das linhas do programa.

Observação: nesta questão só serão consideradas respostas completas.

a)	b)	c)																				
<table border="1"><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																		<table border="1"><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table>				

2) [30 pontos] No jogo de xadrez o tabuleiro pode representado por uma matriz 8 x 8 e cada jogador tem 16 peças brancas ou pretas, sendo que 8 dessas peças são os "peões". No início do jogo, os peões são posicionados um ao lado do outro no tabuleiro e cada peão pode avançar somente na sua coluna.

Faça a função `checaMovimentacao` para verificar se a movimentação do peão é válida. A chamada da função deve ser:
`posicaoOcupada = checaMovimentacao(lin, col, novalin, novacol, tabuleiro)`

onde:

- as variáveis **lin** e **col**, inteiras, fornecem os índices que identificam as coordenadas da posição atual do **peão**;
- as variáveis **novalin** e **novacol**, inteiras, fornecem as coordenadas da nova posição para onde o **peão** deverá ser movido;
- a matriz **tabuleiro**, matriz de inteiros, representa o tabuleiro do jogo e cada um de seus elementos pode armazenar valores entre 0 e 6, onde 0 representa uma posição vazia, 1 indica que a posição é ocupada por um peão e os valores 2 a 6 indicam que a posição é ocupada por outro tipo de peça.

A função `checaMovimentacao` deve retornar 0, caso a nova posição do peão esteja vazia; 1, caso contrário; ou -1 em situações de erro. As situações de erro que devem ser verificadas são: tentar movimentar o peão para uma coluna diferente da sua; tentar avançar o peão em três ou mais casas; tentar retroceder ou não avançar o peão; tentar posicionar o peão fora dos limites de linhas e colunas do tabuleiro.

3) [55 pontos] Utilize a estrutura definida ao lado nas questões a seguir:

```

typedef struct st_produto {
    int codigo;
    char nome[100];
    int vendas[12];
} produto;

```

a) Faça uma função que identifique o produto mais vendido em um ano. Esta função deve **receber** dois parâmetros: um vetor com a lista de produtos (estrutura acima) e o seu tamanho. A função deve **retornar** a posição (índice) na lista do produto mais vendido no ano. Para calcular a quantidade vendida de um produto, considere que o campo `vendas` armazena a quantidade de vendas do produto em cada mês do ano. Por exemplo, a posição 0 armazena a quantidade vendida em janeiro, a posição 3 armazena a quantidade de vendas em abril, etc.

b) Faça um programa principal que **declare** um vetor de produtos, mas considere que a loja possui capacidade para no máximo 100 produtos. O programa deve **fazer a leitura de todos** os produtos disponíveis através do teclado e depois **chamar** a função criada anteriormente para identificar o produto mais vendido. Ao final, o programa deve **imprimir** o nome e o código do produto mais vendido.

c) Crie uma nova estrutura para representar um **fornecedor** de produtos. A estrutura deve conter três campos, o nome do fornecedor, o número de produtos adquiridos deste fornecedor e uma lista de produtos (vetor).