

Lista Ponteiros, Recursão e Struct

1. Dado o Programa a seguir, completar as Tabelas 1 e 2.

```
1 main()
2 {
3     int i, j, *p_1, *p_2, **p_p_1, **p_p_2;
4     i = 4;
5     j = 5;
6     p_1 = &i;
7     p_2 = &j;
8     p_p_1 = &p_2;
9     p_p_2 = &p_1;
10 }
```

Tabela 1

Nome Variável	i	j	p_1	p_2	p_p_1	p_p_2
Conteúdo	4	5				
Endereco	1000	1007	1030	1053	1071	1079

Tabela 2

[illegible]

2. Dado o Programa a seguir, completar as Tabelas 3 e 4.

```
1  main()  
2  {  
3      int i, j, *p_1, *p_2, **p_3, ***p_4;  
4      i = 4;  
5      j = 5;  
6      p_1 = &j;  
7      p_2 = &i;  
8      p_3 = &p_1;  
9      p_4 = &p_3;  
10 }
```

Tabela 3

Nome Variável	i	j	p_1	p_2	p_3	p_4
Conteúdo	4	5				
Endereco	1000	1007	1030	1053	1071	1079

Tabela 4

[illegible]

3. Dado o Programa a seguir, completar as Tabelas 5 e 6.

```
1  main()
2  {
3      int i, j, *p_1, **p_2, **p_3, ***p_4;
4      i = 4;
5      j = 5;
6      p_1 = &j;
7      p_2 = &p_1;
8      p_3 = p_2;
9      p_4 = &p_3;
10 }
```

Tabela 5

Nome Variável	i	j	p_1	p_2	p_3	p_4
Conteúdo	4	5				
Endereço	1000	1007	1030	1053	1071	1079

Tabela 6

[illegible]

4. Dado o Programa a seguir, completar as Tabelas 7 e 8.

```
1 int main()  
2 {  
3     int i, *p_1, *p_2, v[10];  
4     p_1 = v;  
5     p_2 = p_1;  
6     for (i = 0; i < 10; i++) {  
7         v[i] = (2*i)+1;  
8         p_2++;  
9     }  
10    for (i = 0; i < 10; i++) {  
11        p_2--;  
12        printf("[%d]", *p_2);  
13    }
```

Tabela 7 – Valores contidos no vetor v.

[illegible]

Tabela 8 – Valores impressos com: `printf(" [%d] ", *p_2).`

[illegible]

5. Dado o Programa a seguir, completar as Tabelas 9 e 10.

```

1  main()
2  {
3      int i;
4      double *p_1, *p_2, v[10];
5      p_1 = v;
6      p_2 = p_1;
7      for (i = 0; i < 10; i++) {
8          v[i] = i*i+1;
9          p_2++;
10     }
11     for (i = 0; i < 10; i++) {
12         printf(" %d \n", p_2);
13         p_2++;
14     }
15 }

```

Tabela 9 – Valores contidos no vetor v.

Nome Variável	v[0]	v[1]	v[2]	v[3]	v[4]	v[5]	v[6]	v[7]	v[8]	v[9]	v[10]
Conteúdo											
Endereço	5000										

Tabela 10 – Valores impressos com: printf(" [%d] ", p_2)

Valor de i	0	1	2	3	4	5	6	7	8	9	10
p_2											

6. Mostre, através de teste de mesa, o resultado das seguintes funções:

```

(i) int f1(int n)
{
    if (n == 0)
        return (1);
    else
        return(n * f1(n-1));
}

```

Considere as entradas:

- i. f1(0);
- ii. f1(1);
- iii. f1(5);

```

(ii) int f2(int n)
{
    if (n == 0)
        return (1);
    if (n == 1)
        return (1);
    else
        return(f2(n-1) + 2 * f2(n-2));
}

```

Considere as entradas:

- i. $f2(0)$;
- ii. $f2(1)$;
- iii. $f2(5)$;

7. Desenvolva algoritmos recursivos para os seguintes problemas:

- i) Impressão de um número natural em base binária.
- ii) Multiplicação de dois números naturais, através de somas sucessivas (Ex.: $6 * 4 = 4 + 4 + 4 + 4 + 4 + 4$).
- iii) Inversão de uma string.
- iv) Gerador da sequência dada por:
 - $F(1) = 1$
 - $F(2) = 2$
 - $F(n) = 2 * F(n - 1) + 3 * F(n - 2)$.
- v) Faça uma função recursiva que permita inverter um número inteiro N. Ex: 123 – 321
- vi) Crie um programa em C que receba um vetor de números reais com 100 elementos. Escreva uma função recursiva que inverta ordem dos elementos presentes no vetor.

8. Construa uma estrutura aluno com nome, número de matrícula e curso. Leia do usuário informação de 5 alunos, armazene em vetor dessa estrutura e imprima os dados na tela.

9. Crie uma estrutura representando os alunos de um determinado curso. A estrutura deve conter a matrícula do aluno, nome, nota da primeira prova, nota da segunda prova e nota da terceira prova.

- a) Permita ao usuário entrar com os dados de 5 alunos.
- b) Encontre o aluno com maior nota da primeira prova.
- c) Encontre o aluno com maior média geral.
- d) Encontre o aluno com menor média geral.
- e) Para cada aluno diga se ele foi aprovado ou reprovado, considerando o valor 6 para aprovação.