

## Fatec Ipiranga "Pastor Enéas Tognini"

Avaliação P1	Data: _04/_04/_2025_	Nota:
Aluno: <b>Gabarito A</b>	Curso: ADS	
Disciplina: Estrutura de Dados	Turma: 3º semestre - Tarde	

1. (funções e passagem de parâmetro = 2,0) Considere o código a seguir:

```

1  #include <stdio.h>
2  void somaprod (int a, int b, int *c, int *d) {
3      *c = a + b;
4      *d = a * b;
5  }
6  int main () {
7      int soma, prod;
8      somaprod (3, 5, &soma, &prod);
9      printf ("soma = %d, produto = %d", soma, prod);
10     return 0;
11 }
```

a) complete adequadamente a linha 8

b) na tela será impressa a sentença

soma = 8, produto = 15

2. (vetores e funções = 3,0) Considere a busca linear vista em sala:

```

int busca_simples(int x, int v[], int n){
    for (int i = 0; i < n; i++)
        if (v[i] == x)
            return 1;
    return 0;
}
```

Com base na função anterior, desenvolver uma função que tem como entradas um vetor v, seu tamanho n e um elemento x; e saída a quantidade de vezes que o x aparece em v.

```

int vezes (int x, int v[], int n) {
    int cont = 0;
    for (int i = 0; i < n; i++)
        if (v[i] == x)
            cont++;
    return cont;
}
```

# Fatec Ipiranga "Pastor Enéas Tognini"

Avaliação P1	Data: _04_/_04_/_2025_	Nota:
Aluno: Gabarito B	Curso: ADS	
Disciplina: Estrutura de Dados	Turma: 3º semestre - Tarde	

4. (funções e passagem de parâmetro = 2,0) Considere o código a seguir:

```

1 #include <stdio.h>
2 void divmod(int a, int b, int *divisao, int *resto) {
3     *divisao = a / b;
4     *resto = a % b;
5 }
6 int main() {
7     int div, resto;
8     divmod(20, 6, &div, &resto);
9     printf("divisão = %d, resto = %d\n", div, resto);
10    return 0;
11 }
```

a) complete adequadamente a linha 8

b) na tela será impressa a sentença

divisao = 3, resto = 2

5. (vetores e funções = 3,0) Considere a busca linear vista em sala:

```

int busca_simples(int x, int v[], int n){
    for (int i = 0; i < n; i++)
        if (v[i] == x)
            return i;
    return -1;
}
```

Com base na função anterior, desenvolver uma função que tem como entradas um vetor v, seu tamanho n e um elemento x; e saída a posição da primeira ocorrência de x em v. Caso ele não apareça, retornar -1.

```

int primeira (int x, int v[], int n) {
    for (int i = 0; i < n; i++)
        if (v[i] == x)
            return i;
    return -1;
}
```

# Fatec Ipiranga "Pastor Enéas Tognini"

Avaliação P1	Data: 04 / 04 / 2025	Nota:
Aluno: Gubairto A / B	Curso: ADS	
Disciplina: Estrutura de Dados	Turma: 3º semestre - Tarde	

(ordenação = 3,0) O Selection Sort é um algoritmo de ordenação que seleciona o menor elemento de uma sequência e o coloca na primeira posição de um *array*. Esse processo é repetido até que todo o *array* esteja ordenado.

```

1 void selection_sort (int n, int v[]) {
2     for (int i = 0; i < n-1; ++i) {
3         int min = i;
4         for (int j = i+1; j < n; ++j)
5             if (v[j] < v[min])
6                 min = j;
7         int x = v[i];
8         v[i] = v[min];
9         v[min] = x;
10    }
11 }
```

Escreva seu RA aqui

							0	3	8	2	4	1
							v	0	1	2	3	4

Faça o teste de mesa, marcando as variáveis em cada iteração, do `selection_sort`, para o vetor `v` (note que são só as posições de 0 a 4).

i	min	j	x	
0	0	1		
	2	2		
		3		
	4	4		
		5 *	3	$v[0] = v[4]$ $v[4] = 3$ $[1 \ 8 \ 2 \ 4 \ 3]$
1	1	2		
	2	3		
		4		
		5 *	8	$v[1] = v[2]$ $v[2] = 8$ $[1 \ 2 \ 8 \ 4 \ 3]$

# Fatec Ipiranga "Pastor Enéas Tognini"

Avaliação P1	Data: _04/_04/_2025_	Nota:
Aluno: Gabarito A/B continua:	Curso: ADS	
Disciplina: Estrutura de Dados	Turma: 3º semestre - Tarde	

3. (ordenação = 3,0) O Selection Sort é um algoritmo de ordenação que seleciona o menor elemento de uma sequência e o coloca na primeira posição de um array. Esse processo é repetido até que todo o array esteja ordenado.

```

1 void selection_sort (int n, int v[]) {
2     for (int i = 0; i < n-1; ++i) {
3         int min = i;
4         for (int j = i+1; j < n; ++j)
5             if (v[j] < v[min])
6                 min = j;
7         int x = v[i];
8         v[i] = v[min];
9         v[min] = x;
10    }
11 }

```

Escreva seu RA aqui

							0	3	8	2	4	1
							v	0	1	2	3	4

Faça o teste de mesa, marcando as variáveis em cada iteração, do selection\_sort, para o vetor v (note que são só as posições de 0 a 4).

<sup>0</sup> <sup>1</sup> <sup>2</sup> <sup>3</sup> <sup>4</sup>  
[ 1 2 8 4 3 ]

i min j x

2 2 3  
3 4  
4 5 \* 8

v[2] = v[4]

v[4] = 8

[ 1 2 3 4 8 ]

3 3 4  
5 \* 4

v[3] = v[3]

v[3] = 4

[ 1 2 3 4 8 ]

4 \*