



Aluno: \_\_\_\_\_ No. \_\_\_\_\_

A cola não será tolerada. Se alguém for pego colando, será reprovado com Zero. É considerado cola: olhar/copiar da prova de outro ou deixar outro aluno olhar sua prova.

### 1ra. Avaliação - Grupo A

1. (1 pt) Fazer um programa em C que seja capaz de ler número inteiros positivos do teclado e de identificar o maior destes. Ao final, o maior número encontrado deve ser impresso. O número zero é o último número fornecido como dado de entrada. Observação: o zero não deve ser considerado na busca do maior.

```
int main(){
    int num, maior;
    printf("Digite valor (0 para finalizar)");
    scanf("%d", &num);
    maior = num;
    while(num != 0){
        if (num > 0 && num > maior){
            maior = num;
        }
        printf("Digite valor (0 para finalizar)");
        scanf("%d", &num);
    }
    printf("O numero maior e: %d", maior);
    return 0;
}
```

2. (2 pts) Faça um programa em C que leia um número e mostre uma mensagem indicando se este número é par ou ímpar e se é positivo ou negativo.

```
int main(){
    int num;
    printf("Digite um número");
    scanf("%d", & num);
    if (num % 2 == 0)
        printf("Numero Par \n");
    else
        printf("Numero Impar \n");
    if (num > 0)
        printf("Numero Positivo");
    else
        printf("Numero Negativo");
    return 0;
}
```

3. (2 pts) Escreva um programa que permita fazer o pagamento de um funcionário, de acordo com a seguinte tabela

Código	Cargo	Pagamento
1	Presidente	R\$ 20.000,00
2	Diretor	R\$ 10.000,00 + bonus
3	Assistente	R\$ 500,00 + bonus + R\$ 10 / hora
4	Vendedor	R\$ 800,00 + 15% das vendas

O programa deve ler o nome do funcionário, o código correspondente ao seu cargo e indicar o valor a ser recebido. Caso haja dados extras necessários para o cálculo do pagamento, o programa deve perguntar por eles somente se houver necessidade.

Obs.: use somente comandos **switch**.

```
int main(){
    double salario , bonus , vendas , horas;
    int cod;
    char nome[50];
    printf("Digite nome do funcionario: ");
    fgets(nome, 50, stdin);
    printf("Digite codigo: ");
    scanf("%d", cod);
    switch(cod){
        case 1:
            salario = 20000;
            break;
        case 2:
            printf("Inserir valor do bonus: ");
            scanf("%lf", &bonus);
            salario = 10000 + bonus;
            break;
        case 3:
            printf("Inserir valor do bonus: ");
            scanf("%lf", &bonus);
            printf("Inserir numero de horas trabalhadas: ");
            scanf("%lf", &horas);
            salario = 500 + bonus + horas*10;
            break;
        case 4:
            printf("Inserir valor de vendas: ");
            scanf("%lf", &vendas);
            salario = 800 + 0.15 * vendas;
            break;
    }
    printf("%s recebe %lf reais de salario", nome, salario);
    return 0;
}
```

4. (2 pts) Seja a seguinte série:

1, 4, 4, 2, 5, 5, 3, 6, 6, 4, 7, 7, ...

Escreva um programa que seja capaz de gerar os  $N$  termos dessa série. Esse número  $N$  deve ser lido do teclado.

```
int main(){
    int n, i, j, cont;
    printf("Digite numero de termos");
    scanf("%d", &n);
    for(cont = 0, i = 1, j = 4; cont < n; cont++){
        if (cont % 3 == 0)
            printf("%d ", i++);
        if (cont % 3 == 1)
            printf("%d ", j);
        if (cont % 3 == 2)
            printf("%d ", j++);
        }
    return 0;
}
```

5. (2 pts) Fazer um programa que leia um valor ( $X$ ) e depois calcule e escreva o resultado do seguinte somatório

$$\frac{X^{25}}{1} - \frac{X^{24}}{2} + \frac{X^{23}}{3} - \frac{X^{22}}{4} + \dots + \frac{X^1}{25}$$

```
int main(){
    int i, exp;
    double x, s = 0;
    printf("Digite um valor");
    scanf("%lf", &x);
    for (i = 1, exp = 25; i <= 25; i++, exp--){
        s = s + pow(x, exp) / i;
        printf("Resultado = %lf", s);
    }
    return 0;
}
```

6. (2 pts) Considere uma linha ferroviária entre São Paulo e Curitiba. Suponha que uma locomotiva (trem) *A* parte de São Paulo para Curitiba com velocidade de 30 m/s enquanto que uma outra locomotiva *B* parte de Curitiba para São Paulo no mesmo instante com velocidade de 40 m/s. Considere a distância entre São Paulo e Curitiba de 400 Km. Escreva um programa que calcule iterativamente o tempo necessário para os maquinistas pararem as locomotivas antes que uma colisão aconteça. O algoritmo deve calcular também a distância que as locomotivas devem percorrer para que a colisão aconteça.

```
int main(){
    double sp, c;
    int cont = 0;
    sp = 0; c = 400000;
    while(sp < c)
    {
        sp += 30;
        c -= 40;
        cont++;
    }
    printf("passaram %d segundos: \n", cont);
    printf("Trem A deve percorrer %d Km", (cont+1)*30/1000);
    printf("Trem B deve percorrer %d Km", (cont+1)*40/1000);
    return 0;
}
```