



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. Desligar os celulares, quem for pego olhando o celular, será considerado cola. Implementar cada questão utilizando funções e/ou procedimentos.

2da. Avaliação - Grupo A

- (2 pts) Para evitar erros de digitação em números de grande importância, como código de uma conta bancária, geralmente se adiciona ao número um dígito verificador. Por exemplo, o número 1841 é utilizado normalmente como 18414, onde o 4 é o dígito verificador. Ele é calculado da seguinte forma:
 - Cada algarismo do número é multiplicado por um peso começando de 2 da direita para a esquerda. Para cada algarismo o peso é acrescido de 1. Soma-se os produtos obtidos.
$$1 \times 5 + 8 \times 4 + 4 \times 3 + 1 \times 2 = 51$$
 - Calcula-se o resto da divisão desta soma por 11: $51 \% 11 = 7$
 - Subtrai-se de 11 o resto obtido: $11 - 7 = 4$
 - Se o valor obtido for 10 ou 11, o dígito verificador será o 0, nos outros casos, o dígito verificador é o próprio valor encontrado.

Escreva um(a) procedimento/função que gere o dígito verificador de um número (de 1 a 999).

```
int codigo(int num)
{
    int x = 2, soma = 0, r, s;
    if (num < 1000)
    {
        while (num > 0)
        {
            soma += (num % 10)*x;
            x++;
            num = num / 10;
        }
        r = soma % 11;
        s = 11 - r;
        if (s == 10 || s == 11) return 0;
        else return s;
    }
    return -1;
}
```

2. (2 pts) Dado um vetor de 15 números inteiros, foi feito um programa que comprime um vetor suprimindo as repetições de números vizinhos através da contagem do número de repetições de cada um da seguinte forma:

Vetor de entrada:

1 1 1 4 1 1 4 4 25 67 67 67 67 2 2

Vetor de saída:

3 1 1 4 2 1 2 4 1 25 4 67 2 2 0

Faça um programa que descomprima o vetor de saída, gerando o vetor de entrada correspondente.

```
int Decomprime(int *x, int tam, int* v)
{
    int i, j = 0, k;
    for (i = 0; i < tam; i += 2)
    {
        k = 0;
        while (k < x[i])
        {
            v[j++] = x[i + 1];
            k++;
        }
    }
    return j;
}
```

3. (2 pts) Implemente um programa que receba uma linha de texto, retire os espaços em excesso existentes deixando apenas um espaço entre as várias palavras. A linha de texto final não pode começar, nem finalizar com um espaço em branco.

```
void EliminaBranco(char* cad)
{
    int i = 0, j;
    while (cad[i] == ' ')
    {
        i++;
    }
    for (j = 0; i > 0 && j < strlen(cad) - i; j++)
        cad[j] = cad[j + i];
    for (i = 0; i < strlen(cad); i++)
    {
        if (cad[i] == ' ' && cad[i+1] == ' ')
        {
            for (j = i; j < strlen(cad); j++)
                cad[j] = cad[j + 1];
            if (cad[i] == ' ') i--;
        }
    }
    if (cad[strlen(cad) - 1] == ' ')
        cad[strlen(cad) - 1] = '\0';
    for (i = 0; i < strlen(cad); i++)
        printf("%d-%c\n", i, cad[i]);
}
```

4. (2 pts) Na teoria dos sistemas define-se como elemento minimax de uma matriz o menor elemento da linha onde se encontra o maior elemento da matriz. Sabendo disso, escreva um(a) procedimento/função que recebe uma matriz e que determina o elemento minimax e a posição (linha e coluna) que o mesmo ocupa na matriz.

```

#define N 5
void Minimax(int M[N][N], int pos[2])
{
    int max = M[0][0], lin = 0, i, j, min, col;
    // Encontra o maior elemento na matriz
    for (i = 0; i < N; i++)
    {
        for (j = 0; j < N; j++)
        {
            if (M[i][j] > max)
            {
                max = M[i][j];
                lin = i;
            }
        }
    }
    // Encontra o menor da linha lin
    min = M[lin][0];
    col = 0;
    for (j = 1; j < N; j++)
    {
        if (M[lin][j] < min)
        {
            min = M[lin][j];
            col = j;
        }
    }
    pos[0] = lin;
    pos[1] = col;
}

```

5. (2 pts) Leia um vetor de 16 posições e troque os 8 primeiros valores pelos 8 últimos e vice-versa. Escreva ao final o vetor obtido

```

void trocaMetada(int* V, int n)
{
    int i, tmp;
    if (n % 2 == 0)
    {
        for (i = 0; i < n / 2; i++)
        {
            tmp = V[i];
            V[i] = V[i + n / 2];
            V[i + n / 2] = tmp;
        }
    }
    else
    {
        for (i = 0; i < n / 2; i++)
        {
            tmp = V[i];
            V[i] = V[i + n / 2 + 1];
            V[i + n / 2 + 1] = tmp;
        }
    }
}

```