



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.

2da. Avaliação - Grupo A

1. (2pts) Elaborar um procedimento que receba uma matriz A de duas dimensões com cinquenta linhas e cinquenta colunas. Calcular o somatório dos elementos situados nas posições de linha e coluna ímpares da diagonal principal da referida matriz.

```
int SomaDiagonalImpares(int M[50][50], int lin, int col)
{
    int i, soma = 0;
    for (i = 1; i < lin; i+=2)
        soma += M[i][i];
    return soma;
}
int main(){
    int Z[50][50], i, j;
    for (i = 0; i < 50; i++){
        for (j = 0; j < 50; j++){
            printf("Digite um numero: ");
            scanf("%d", &Z[i][j]);
        }
        printf("Soma das posicoes impares da diagonal %d",
            SomaDiagonalImpares(Z, 50, 50) );
    }
    return 0;
}
```

2. (2pts) Fazer um procedimento que receba uma *string* e um caracter qualquer e construir um outro vetor (*OC*) contendo as posições (índices) de onde ocorre o caracter na *string*. Exemplo: Seja a string “abracadabra!!!” e o caracter ‘a’, então o vetor de índices deverá conter os seguintes valores: 0 3 5 7 10 -1. O valor -1 indica final de vetor, ou seja, que não existem mais ocorrências. Caso, não exista nenhuma ocorrência, deve ser armazenado o valor -1 no vetor. No exemplo acima se o caracter fosse ‘x’, *OC* seria -1.

```
void Posicoes(char* cad, char car, int* OC, int tam)
{
    int i, j;
    for (i = 0, j = 0; i < strlen(cad); i++)
        if (cad[i] == car)
            OC[j++] = i;
    OC[j] = -1;
}
int main(){
    char cadeia[50], letra;
```

```

int OC[50];
printf("Inserir cadeia: ");
fgets(cadeia, 50, stdin);
printf("Inserir caracter");
scanf("%c", &letra);
Posicoes(cadeia, letra, OC);
printf("Mostrando as posicoes que ocupa o caracter %c", letra);
for (i = 0; OC[i] != -1 && i < 50; i++)
    printf("%d ", OC[i]);
return 0;
}

```

3. (2pts) Escreva um procedimento que receba uma *string* correspondendo ao nome de uma pessoa e crie uma nova *string* contendo apenas as iniciais do nome. (Ex: “João da Silva” => “JdS”)

```

void Iniciais(char* cad1, char* cad2)
{
    int i, j = 0;
    char copia[100] = " ";
    strcat(copia, cad1);
    for (i = 0; i < strlen(copia)-1; i++)
        if (copia[i] == ' ' && copia[i+1] != ' ')
            cad2[j++] = copia[i+1];
    cad2[j] = '\0';
}
int main(){
    char cad[50], ini[10];
    printf("Digite um nome completo: ");
    fgets(cad, 50, stdin);
    Iniciais(cad, ini);
    printf("As iniciais de %s sao %s ", cad, ini);
    return 0;
}

```

4. (2pts) Dada uma seqüência de N números, determinar quantos números distintos compõem a seqüência e o número de vezes que cada um deles ocorre na mesma. Implemente o procedimento. Exemplo: $N = 5$

```

1 2 3 2 3
a seqüência tem três números distintos, 1, 2 e 3. Ocorrências:
1 1 vez
2 2 vezes
3 2 vezes

```

```

int Ocorrencias(int* V, int* SemRep, int* Vezes, int n)
{
    int i, tot = 0, j;
    // Gera o vetor SemRep sem elementos repetidos
    for (i = 0; i < n; i++){
        for (j = 0; j < tot; j++)
            if (SemRep[j] == V[i])
                break;
        if (j == tot)
            SemRep[tot++] = V[i];
    }
    // Conta o numero de vezes que aparece cada elemento
    for (j = 0; j < tot; j++)
    {
        Vezes[j] = 0;
        for (i = 0; i < n; i++)

```

```

        if (SemRep[j] == V[i])
            Vezes[j]++;
    }
    return tot;
}
int main(){
    int V[5] = {1, 2, 2, 2, 1}, Elem[5], Vezes[5], i, tot;
    tot = Ocorrencias(V, Elem, Vezes, 5);
    for(i = 0; i < tot; i++)
        printf("%d %d \n", Elem[i], Vezes[i]);
    return 0;
}

```

5. (2pts) Implemente o procedimento que receba uma string, retirando-lhe todos os caracteres que não se encontram repetidos.

```

"alface"          ->  "aa"
"alta fidelidade" ->  "alaidelidade"

```

```

void EliminaNaoRep(char* cad, char* Rep)
{
    int n = strlen(cad), i, j, cont, k = 0;
    for (i = 0; i < n; i++){
        cont = 0;
        for (j = 0; j < n; j++)
            if (cad[i] == cad[j])
                cont++;
        if (cont > 1)
            Rep[k++] = cad[i];
    }
    Rep[k] = '\0';
}
int main(){
    char cad[30];
    EliminaNaoRep("alface", cad);
    printf("%s", cad);
    return 0;
}

```