

SSC301 - INTRODUÇÃO À COMPUTAÇÃO PARA A ENGENHARIA AMBIENTAL

Profa. Dra. Milena Guessi Margarido

TRABALHO 01

Dupla:

- João Victor de Araújo Lima
- Hingryd Lima Rauen

Integrantes do curso de Licenciatura em Ciências Exatas – IFSC/USP

Alunos matriculados na disciplina através do oferecimento da Ênfase em Computação, pelo ICMC em 2017/2.

1) O sistema de avaliação de uma disciplina obedece aos seguintes critérios:

- Durante o semestre são dadas três notas;
- A nota final é obtida pela média aritmética das notas dadas durante o curso;
- É considerado aprovado o aluno que obtiver nota final superior ou igual a 6 e que tiver comparecido a um mínimo de 70% das aulas.

Escreva um programa que:

a) Leia um conjunto de dados contendo o número de matrícula, as três notas (0 a 10) e a frequência de (número de aulas frequentadas) de 100 alunos;

b) Calcule:

- a. A nota final de cada aluno;
- b. A maior e a menor nota da turma;
- c. A nota média da turma;
- d. O total de alunos reprovados;
- e. A porcentagem de alunos reprovados por faltas

c) Imprima na tela:

- a. Para cada aluno, o número de matrícula, a frequência, a nota final e o resultado (aprovado ou reprovado);
- b. O que foi calculado no item b.

Printscreen da tela de saída:

```
"C:\Users\Hingryd\Desktop\Exercicio 1-1.exe"
Programa de ajuda ao professor com a disciplina.
Digite aqui qual o total de aulas dadas durante o periodo do curso: 50
Digite o numero de matricula do 1 aluno: 12
Digite a 1 nota do aluno 12: 4
Digite a 2 nota do aluno 12: 6
Digite a 3 nota do aluno 12: 8
Coloque a quantidade de aulas frequentadas pelo aluno 12: 70
Quantidade de aulas frequentadas invalida. Digite novamente a quantidade de aulas frequentadas pelo aluno 12: 45
Digite o numero de matricula do 2 aluno: 23
Digite a 1 nota do aluno 23: 3
Digite a 2 nota do aluno 23: 1
Digite a 3 nota do aluno 23: 2
Coloque a quantidade de aulas frequentadas pelo aluno 23: 34
Digite o numero de matricula do 3 aluno: 222
Digite a 1 nota do aluno 222: 8
Digite a 2 nota do aluno 222: 9
Digite a 3 nota do aluno 222: 10
Coloque a quantidade de aulas frequentadas pelo aluno 222: 50
Digite o numero de matricula do 4 aluno: 3
Digite a 1 nota do aluno 3: 4
Digite a 2 nota do aluno 3: 5
Digite a 3 nota do aluno 3: 6
Coloque a quantidade de aulas frequentadas pelo aluno 3: 34
Digite o numero de matricula do 5 aluno: 4444
Digite a 1 nota do aluno 4444: 1
Digite a 2 nota do aluno 4444: 2
Digite a 3 nota do aluno 4444: 3
Coloque a quantidade de aulas frequentadas pelo aluno 4444: 4
Digite o numero de matricula do 6 aluno: 8
Digite a 1 nota do aluno 8: 6
Digite a 2 nota do aluno 8: 7
Digite a 3 nota do aluno 8: 5
Coloque a quantidade de aulas frequentadas pelo aluno 8: 50
Digite o numero de matricula do 7 aluno: 8
Digite a 1 nota do aluno 8: 0
Digite a 2 nota do aluno 8: 0
Digite a 3 nota do aluno 8: 1
Coloque a quantidade de aulas frequentadas pelo aluno 8: 3
Digite o numero de matricula do 8 aluno: 4
Digite a 1 nota do aluno 4: 1
Digite a 2 nota do aluno 4: 2
Digite a 3 nota do aluno 4: 3
Coloque a quantidade de aulas frequentadas pelo aluno 4: 44
Digite o numero de matricula do 9 aluno: 1
```

Programa em C (com código comentado – pseudocódigo)

```
#include<stdio.h>

int main () {
    int numaulas; //Variavel criada para termos um controle depois do numero de aulas para
    obter 70% de frequencia
    float minaulas; //numero minimo de aulas para se obter 70% de frequencia
    int matricula[100]; //vetor para receber os numeros de matriculas dos alunos
    float nota1[100]; //vetor para receber as primeiras notas dos alunos
    float nota2[100]; //vetor para receber as segundas notas dos alunos
    float nota3[100]; //vetor para receber as terceiras notas dos alunos
    int frequencia[100]; //vetor para receber os numeros de aulas frequentadas pelos alunos
    float notafinal[100]; //vetor para armazenar as notas finais dos alunos
    int RN = 0; //variavel para saber quantos alunos reprovaram por nota
    int RF = 0; //variavel para saber quantos alunos reprovaram por falta
    int RA = 0; //variavel para saber quantos alunos reprovaram por nota e falta
    int i;
    int recado[100] = {0}; //vetor para printar, no final, se o aluno reprovou ou nao
    (0=nao,1=sim)
    float maiornota = 0.0; //variavel criada para obter maior nota da turma
    float menornota = 10.0; //variavel criada para obter a menor nota da turma
    float somanota = 0.0; //variavel criada para armazenar as notas dos alunos da turma e
    calcular a media da turma posteriormente
    int totalreprovados = 0; //variavel criada para armazenar quantos alunos reprovaram da
    turma

    printf("Programa de ajuda ao professor com a disciplina. \nDigite aqui qual o total de
    aulas dadas durante o periodo do curso: ");
    scanf("%d", &numaulas);
    while (numaulas < 0){
        printf("Numero de aulas dadas invalido.\nDigite novamente o total de aulas dadas
        durante o periodo do curso: ");
        scanf("%d", &numaulas);
    }

    minaulas = (numaulas * 0.7);

    for (i = 0; i < 100; i++){
        printf("Digite o numero de matricula do %d aluno: ", i+1);
        scanf("%d", &matricula[i]);

        printf("Digite a 1 nota do aluno %d: ", matricula[i]);
        scanf("%f", &nota1[i]);
        while((nota1[i]<0)||((nota1[i]>10))){
```

```

        printf("Nota invalida. Digite novamente a 1 nota do aluno %d: ", matricula[i]);
        scanf("%f", &nota1[i]);
    }

    printf("Digite a 2 nota do aluno %d: ", matricula[i]);
    scanf("%f", &nota2[i]);
    while((nota2[i]<0)|| (nota2[i]>10)){
        printf("Nota invalida. Digite novamente a 2 nota do aluno %d: ", matricula[i]);
        scanf("%f", &nota2[i]);
    }

    printf("Digite a 3 nota do aluno %d: ", matricula[i]);
    scanf("%f", &nota3[i]);
    while((nota3[i]<0)|| (nota3[i]>10)){
        printf("Nota invalida. Digite novamente a 3 nota do aluno %d: ", matricula[i]);
        scanf("%f", &nota3[i]);
    }

    printf("Coloque a quantidade de aulas frequentadas pelo aluno %d: ", matricula[i]);
    scanf("%d", &frequencia[i]);
    while((frequencia[i]<0)|| (frequencia[i]>numaulas)){
        printf("Quantidade de aulas frequentadas invalida. Digite novamente a quantidade
de aulas frequentadas pelo aluno %d: ", matricula[i]);
        scanf("%d", &frequencia[i]);
    }

    notafinal[i] = (nota1[i] + nota2[i] + nota3[i]) / 3.0;
    if (notafinal[i] < 6){
        RN++;
        recado[i] = 1;
    }

    if (frequencia[i] < minaulas){
        RF++;
        recado[i] = 1;
    }

    if ((notafinal[i] < 6) && (frequencia[i] < minaulas)){
        RA++;
    }

    if (notafinal[i] > maiornota){
        maiornota = notafinal[i];
    }

    if (notafinal[i] < menornota){
        menornota = notafinal[i];
    }

```

```

        if (recado[i] == 1){
            totalreprovados++;
        }
        somanota += notafinal[i];
    }

    //fazer média com somanota no printf
    //criar variavel para medir frequencia por porcentagem

    printf("Matricula\tFrequencia\tNota\tResultado\n");
    for (i = 0; i < 100; i++){
        if (recado[i] == 0){
            printf("%d\t%d\t%.2f\tAprovado\n", matricula[i], frequencia[i], notafinal[i]);
        }
        else{
            printf("%d\t%d\t%.2f\tReprovado", matricula[i], frequencia[i], notafinal[i]);
        }
    }

    //printar maior e menor nota da turma
    //printar nota média da turma
    //printar o total de alunos reprovados (e seus motivos)
    //printar a porcentagem de alunos reprovados por falta

    return 0;
}

```

2) Fazer um programa que:

- Leia uma matriz inteira A de M x N, onde os elementos de cada linha e os valores de M e N são fornecidos ($M \leq 20$ e $N \leq 10$)
- Imprima a matriz lida
- Calcule e imprima uma matriz modificada B ($M \times N+1$), sendo que os elementos da coluna N+1 são formados com o produto dos elementos da mesma linha.

Exemplo:

A

2	3
4	5

B

2	3	6
4	5	20

Printscreen da tela de saída do exercício 02:

```
"I:\Enfase\trabalho 01\exercicio02.exe"
Programa para trabalhar com matrizes.
Inicialmente pegaremos a quantidade de colunas e linhas tera sua matriz A.
Linhas (entre 0 e 20):2
Colunas (entre 0 e 10): 2
Digite um valor para a linha 0 e coluna 0:1
Digite um valor para a linha 0 e coluna 1:2
Digite um valor para a linha 0 e coluna 2:1
Digite um valor para a linha 1 e coluna 0:3
Digite um valor para a linha 1 e coluna 1:2
Digite um valor para a linha 1 e coluna 2:6
Digite um valor para a linha 2 e coluna 0:4
Digite um valor para a linha 2 e coluna 1:3
Digite um valor para a linha 2 e coluna 2:8
A matriz A(MxN) lida eh:
1      2      1
3      2      6
4      3      8
A matriz B(MxN+1) eh:
1      2      1      2      3      2      6      36      4      3      8      96      Press any key to continue . . .
```

Programa em C:

```
#include<stdio.h>

int main (){
    int A[20][10];
    int B[20][11];
    int M, N;
    int i, j;
    int temp;

    printf("Programa para trabalhar com matrizes.\nInicialmente pegaremos a quantidade de
colunas e linhas tera sua matriz A.\nLinhas (entre 0 e 20): ");
    scanf("%d", &M);
    while((M<0)|| (M>20)){
        printf("Opcao invalida. Digite um numero de linhas entre 0 e 20: ");
        scanf("%d", &M);
    }

    printf("Colunas (entre 0 e 10): ");
    scanf("%d", &N);
    while((N<0)|| (N>10)){
        printf("Opcao invalida. Digite um numero de colunas entre 0 e 10: ");
        scanf("%d", &N);
    }

    for(i=0; i<=M; i++){
        for(j=0; j<=N; j++){
            printf("Digite um valor para a linha %d e coluna %d: ", i, j);
            scanf("%d", &A[i][j]);
        }
    }

    printf("A matriz A(MxN) lida eh:\n");

    for(i=0; i<=M; i++){
        for(j=0; j<=N; j++){
            printf("%d\t", A[i][j]);
        }
        printf("\n");
    }

    printf("A matriz B(MxN+1) eh:\n");

    for(i=0; i<=M; i++){
        temp = 1;
        for(j=0; j<=(N+1); j++){
```

```
        if(j<=N){
            B[i][j] = A[i][j];
            printf("%d\t", B[i][j]);
            temp *= A[i][j]
        }
        else{
            B[i][j] = temp;
            printf("%d\t", B[i][j]);
        }
    }
}

system("pause");

return 0;
}
```


3) Intercalação é o processo utilizado para construir uma lista ordenada, de tamanho $n+m$, a partir de duas listas já ordenadas de tamanhos n e m . Por exemplo, a partir das listas:

A

1	3	6	7
---	---	---	---

B

2	4	5
---	---	---

Construímos a lista C

1	2	3	4	5	6	7
---	---	---	---	---	---	---

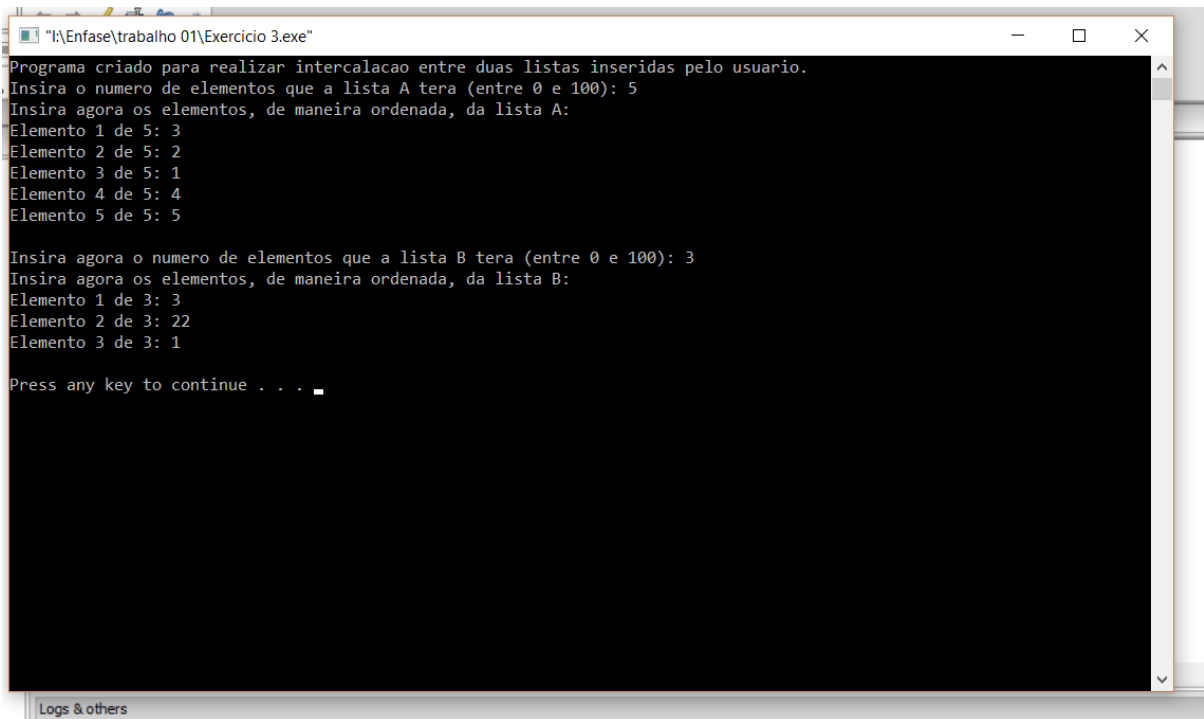
Fazer um programa que:

- Leia NA , número de elementos do conjunto A ($NA \leq 100$);
- Leia, em seguida, os elementos do conjunto A;
- Leia, logo após o valor de NB , número de elementos do conjunto B ($NB \leq 100$);
- Leia, finalmente, os elementos do conjunto B;
- Crie e imprima um conjunto C, ordenado, de tamanho $NA+NB$, a partir dos conjuntos originais A e B.

Observações:

- Considerar os elementos de A e B como inteiros
 - Considerar que os elementos de A e B já são lidos ordenados
-

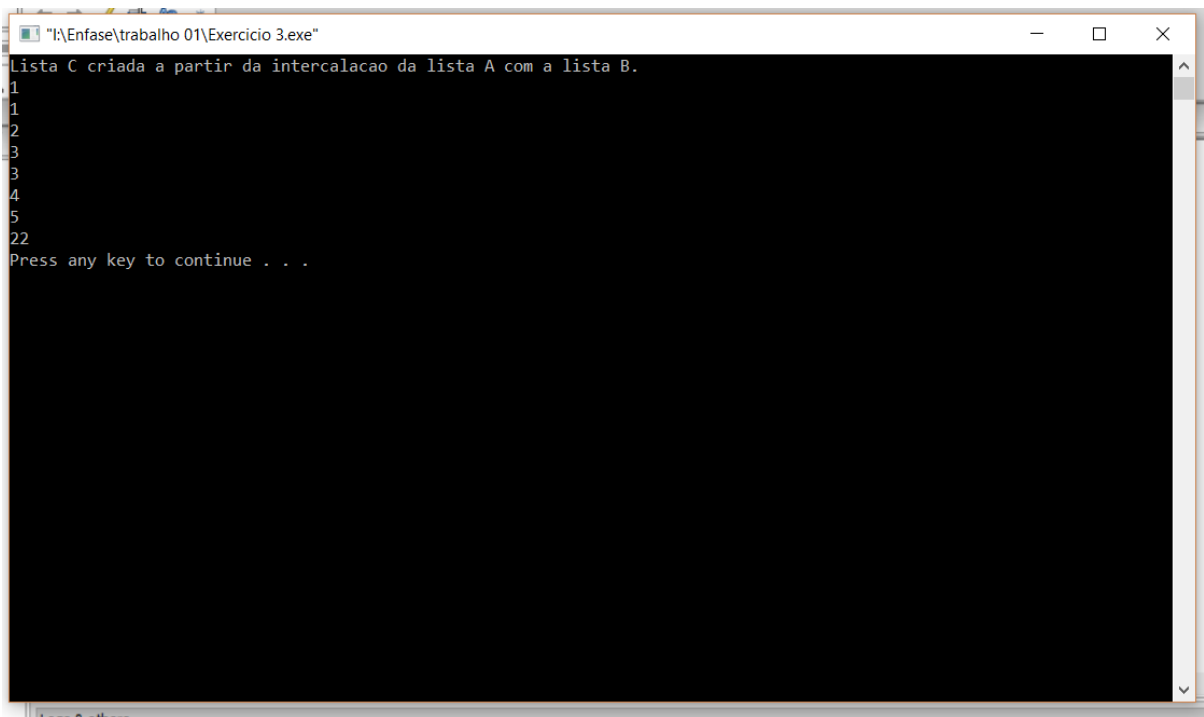
Printscreen da tela de saída



```
"I:\Enfase\trabalho 01\Exercicio 3.exe"
Programa criado para realizar intercalacao entre duas listas inseridas pelo usuario.
Insira o numero de elementos que a lista A tera (entre 0 e 100): 5
Insira agora os elementos, de maneira ordenada, da lista A:
Elemento 1 de 5: 3
Elemento 2 de 5: 2
Elemento 3 de 5: 1
Elemento 4 de 5: 4
Elemento 5 de 5: 5

Insira agora o numero de elementos que a lista B tera (entre 0 e 100): 3
Insira agora os elementos, de maneira ordenada, da lista B:
Elemento 1 de 3: 3
Elemento 2 de 3: 22
Elemento 3 de 3: 1

Press any key to continue . . .
```



```
"I:\Enfase\trabalho 01\Exercicio 3.exe"
Lista C criada a partir da intercalacao da lista A com a lista B.
1
1
2
3
3
4
5
22
Press any key to continue . . .
```

Programa em C:

```
#include<stdio.h>
```

```
int main (){
```

```
    int NA;
```

```
    int NB;
```

```
    int A[100];
```

```
    int B[100];
```

```
    int C[200];
```

```
    int i, j;
```

```
    int temp;
```

```
    printf("Programa criado para realizar intercalacao entre duas listas inseridas pelo  
usuario.\nInsira o numero de elementos que a lista A tera (entre 0 e 100): ");
```

```
    scanf("%d", &NA);
```

```
    while ((NA<0)|| (NA>100)){
```

```
        printf("Valor invalido para o numero de elementos de A.\n Insira novamente o numero  
de elementos da lista A: ");
```

```
        scanf("%d", &NA);
```

```
    }
```

```
    printf("Insira agora os elementos, de maneira ordenada, da lista A:\n");
```

```
    for(i=0; i<NA; i++){
```

```
        printf("Elemento %d de %d: ", (i+1), NA);
```

```
        scanf("%d", &A[i]);
```

```
    }
```

```
    printf("\n");
```

```
    printf("Insira agora o numero de elementos que a lista B tera (entre 0 e 100): ");
```

```
    scanf("%d", &NB);
```

```
    while ((NB<0)|| (NB>100)){
```

```
        printf("Valor invalido para o numero de elementos de B.\n Insira novamente o numero  
de elementos da lista B: ");
```

```
        scanf("%d", &NB);
```

```
    }
```

```
    printf("Insira agora os elementos, de maneira ordenada, da lista B:\n");
```

```
    for(i=0; i<NB; i++){
```

```
        printf("Elemento %d de %d: ", (i+1), NB);
```

```
        scanf("%d", &B[i]);
```

```
    }
```

```
    printf("\n");
```

```
    system("pause");
```

```
    system("cls");
```

```

//g=0;

printf("Lista C criada a partir da intercalacao da lista A com a lista B.\n");

for(i=0; i<NA; i++){
    C[i] = A[i];
}
for(i=0; i<NB; i++){
    C[(NA+i)] = B[i];
}

for(i=0; i<(NA+NB); i++){
    for(j=i; j<(NA+NB); j++){
        if(C[j]<C[i]){
            temp=C[i];
            C[i]=C[j];
            C[j]=temp;
        }
    }
}

for(i=0; i<(NA+NB); i++){
    printf("%d\n", C[i]);
}

system("pause");

return 0;
}

```