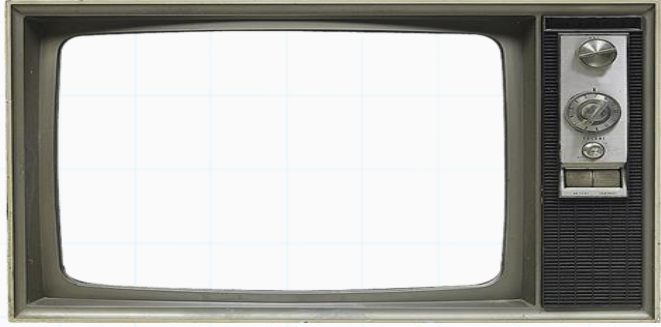
Programação De Computadores



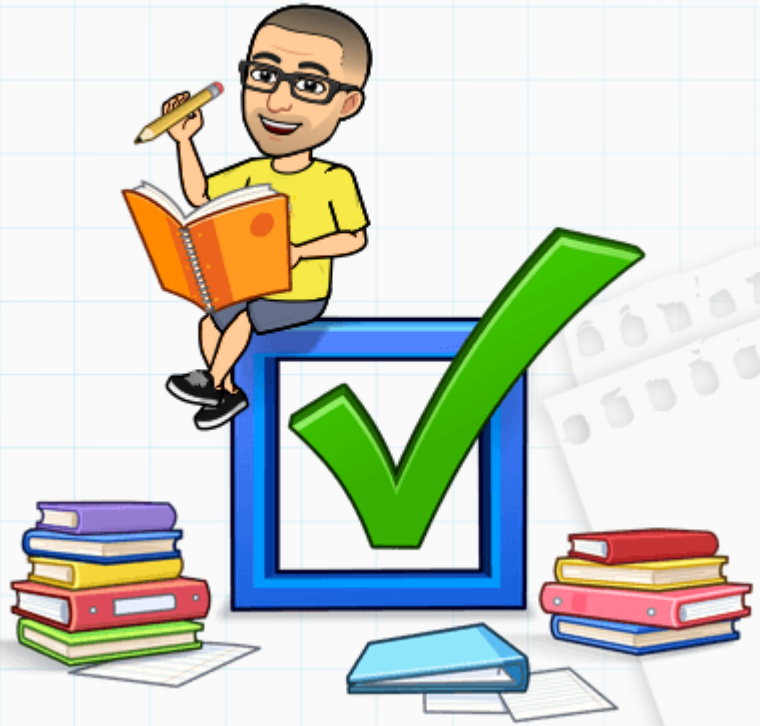
Professor : Yuri Frota

www.ic.uff.br/~yuri/prog.html

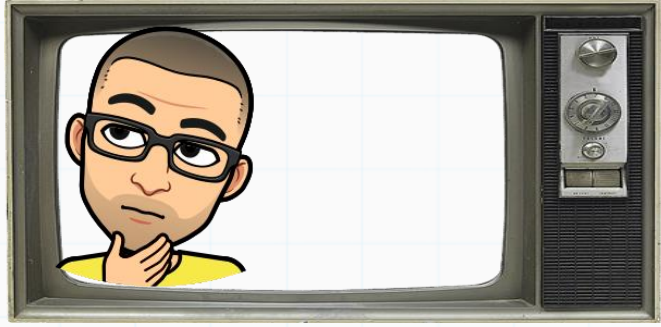
yuri@ic.uff.br

Usar apenas comandos de listas vistos na aula:

- **append (inserir elemento no final da lista)**
- **len (tamanho da lista)**
- **+ (concatenação de listas)**
- *** (multiplicação de listas)**



Listas - LAB



Exercício 1): A coordenação do curso de computação deseja saber quantos alunos estão cursando (de forma irregular) as disciplinas de PROG1, PROG2 e PROG3 ao mesmo tempo. Faça um programa que

(i) leia as matriculas (inteiro) dos 5 alunos de PROG1, dos 7 alunos de PROG2 e 7 alunos de PROG3,

(ii) imprima as matrículas dos alunos irregulares (que fazem as três disciplinas) e

(iii) imprima no fim a quantidade de alunos irregulares.

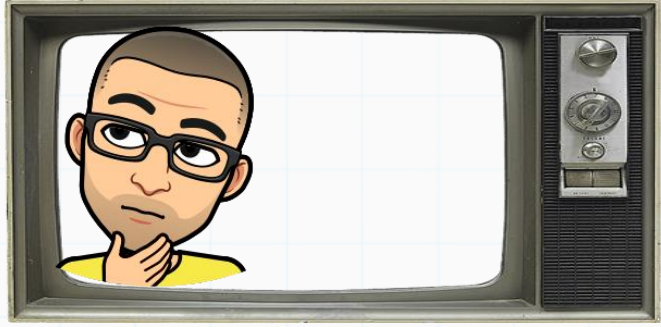
Ex. Execução:

```
PROG1
0) 23
1) 47
2) 12
3) 8
4) 7
PROG2
0) 34
1) 8
2) 23
3) 76
4) 82
5) 7
6) 47
PROG3
0) 12
1) 34
2) 32
3) 23
4) 99
5) 7
6) 76
Aluno  23  irregular
Aluno  7  irregular
total = 2
```

DICAS
Na próxima página

código

Listas - LAB

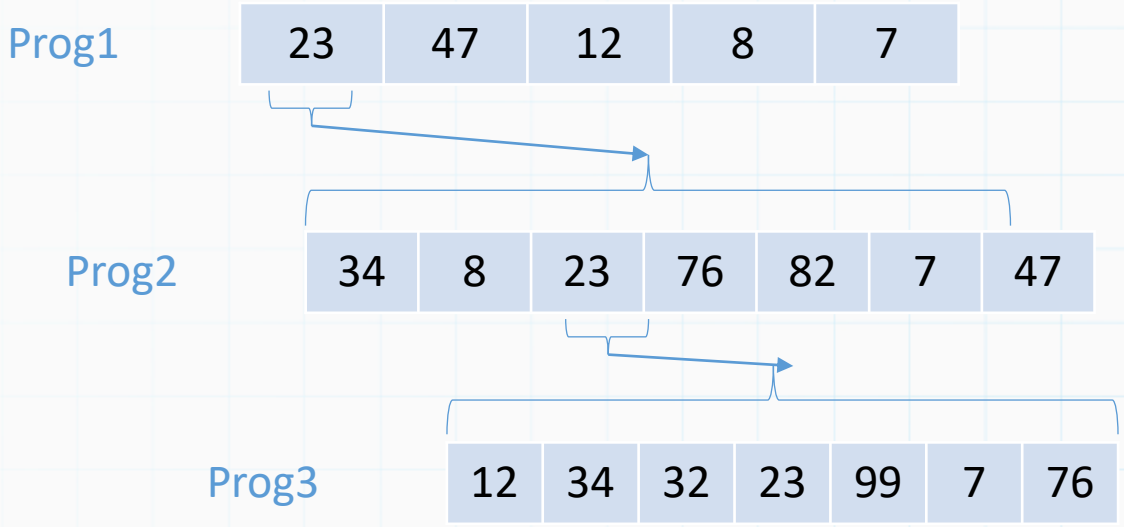


Exercício 1): A coordenação do curso de computação deseja saber quantos alunos estão cursando (de forma irregular) as disciplinas de PROG1, PROG2 e PROG3 ao mesmo tempo. Faça um programa que

(i) leia as matriculas (inteiro) dos 5 alunos de PROG1, dos 7 alunos de PROG2 e 7 alunos de PROG3,

(ii) imprima as matrículas dos alunos irregulares (que fazem as três disciplinas) e

(iii) imprima no fim a quantidade de alunos irregulares.




```
Aluno  23  irregular
Aluno  7  irregular
total = 2
```

Dica: Essa questão pode ser feita com laco triplo (isto eh, 3 níveis de laco)

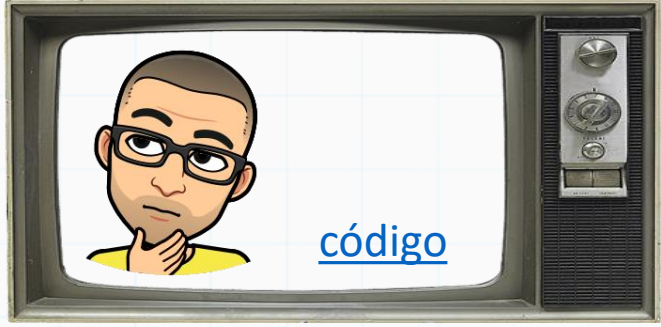
I percorre o vetor PROG1

J percorre o vetor PROG2

K percorre o vetor PROG3

Se o I ésimo aluno de PROG1 for o mesmo que o J ésimo de PROG2 e o mesmo do K ésimo de PROG3 então irregular

Listas - LAB



Exercício 2): Faça um programa que leia um vetor **vet** de 20 números inteiros. O programa deve gerar, a partir do vetor lido, um outro vetor **pos** que contenha apenas os valores inteiros positivos de **vet**. A partir do vetor pos, deve ser gerado um outro vetor **semrep** que contenha apenas uma ocorrência de cada valor de **pos** .
Ex. de execução:

```
0) -2
1) 3
2) 4
3) -5
4) 3
5) 1
6) 1
7) -9
8) 2
9) 13
vet = [-2, 3, 4, -5, 3, 1, 1, -9, 2, 13]
pos = [3, 4, 3, 1, 1, 2, 13]
semrep = [3, 4, 1, 2, 13]
```

Dica do **pos**: Comece **pos** vazio, percorra o **vetor** original e insira (append) no vetor **pos** apenas elementos que forem positivos**.**

Dica **semrep**: Comece **semrep** vazio, e percorra o vetor **pos**, e para cada elemento de **pos** temos que checar se já está em **semrep** (percorrendo o semrep), se não tiver, o inserimos (append) em **semrep.**

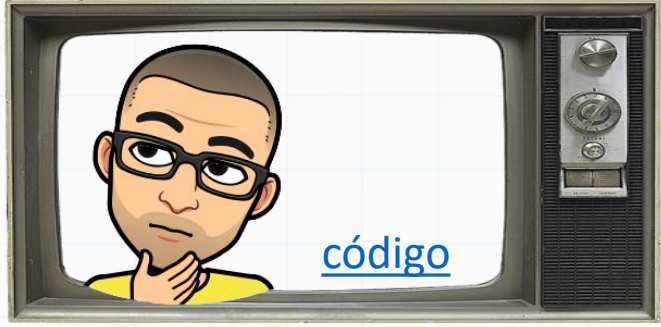
I percorre o vetor POS

J percorre o vetor SEMREP

Testa se o I ésimo número do POS está no SEMREP ?

Se o I ésimo número do POS não está no SEMREP então insere no SEMREP

Listas - LAB



Exercício 2.5): Escreva um programa que, leia um valor inteiro positivo N, e receba um vetor de inteiros A de tamanho N. Depois gere um vetor B de tamanho N que contenha os elementos de A, porem invertidos:

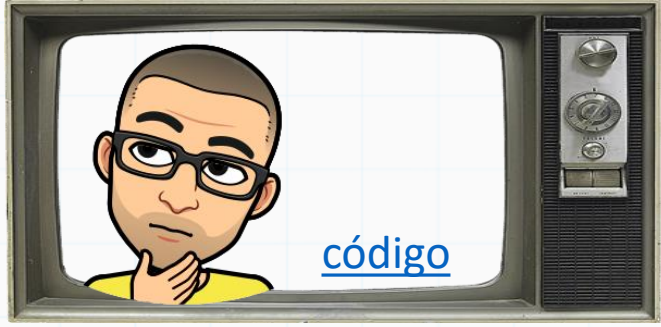
A	5	3	7	1	4
B	4	1	7	3	5

Dica: Comece o vetor B zerado (N zeros). Percorra vetor A (do inicio ao fim), atribuindo o elemento i de A, ao elemento (N-1-i) de B. Perceba que o vetor B vai ser preenchido do fim para o começo.

I percorre o vetor A

(N-1-i)	(i)
B[N-1] =	A[0]
B[N-2] =	A[1]
B[N-3] =	A[2]
.....	

Listas - LAB



Exercício 2.5): Escreva um programa que, leia um valor inteiro positivo N, e receba um vetor de inteiros A de tamanho N. Depois gere um vetor B de tamanho N que contenha os elementos de A, porem invertidos. Agora gere vetor C de tamanho N, com o somatório de cada elemento de B, veja exemplo:

A	5	3	7	1	4
B	4	1	7	3	5
C	10	1	28	6	15

Pois :

4 + 3 + 2 + 1 = 10

1 = 1

7 + 6 + 5 + 4 + 3 + 2 + 1 = 28

3 + 2 + 1 = 6

5 + 4 + 3 + 2 + 1 = 15

I percorre o vetor B

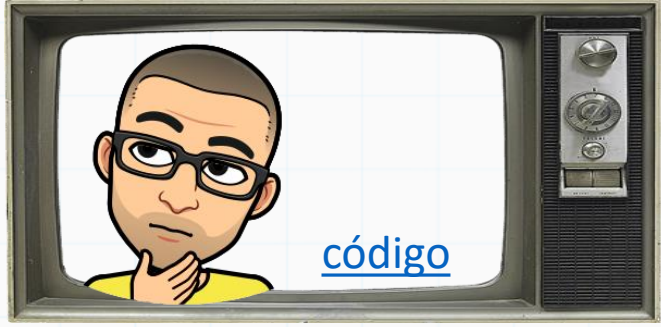
J vai até o I ésimo número de B

Calcula somatório do I ésimo número de B

Insere somatório em C

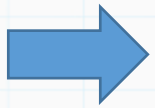
Dica: Comece o vetor C vazio. Percorre elementos de B, para cada elemento, calcula seu somatório e depois insere no vetor C

Listas - LAB



Exercício 3): Escreva um programa que, leia um valor inteiro positivo N, depois receba primeiro um vetor A de tamanho N, depois um vetor B de tamanho N e depois um vetor C de tamanho N. Depois, construa um vetor D de tamanho 3*N, alternando os elementos de A,B e C:

A	B	C
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
...	...	...
a_{N-1}	b_{N-1}	c_{N-1}



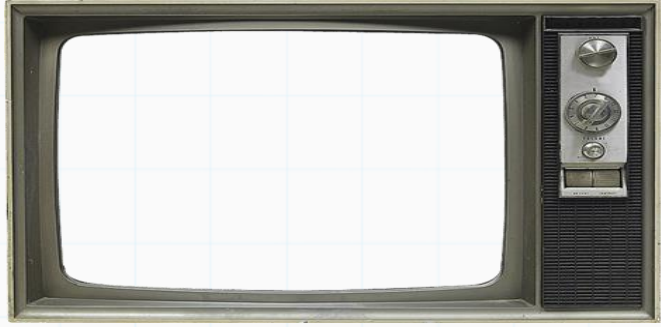
D
a_0
b_0
c_0
a_1
b_1
c_1
a_2
b_2
c_2
...

Ex. de Execução

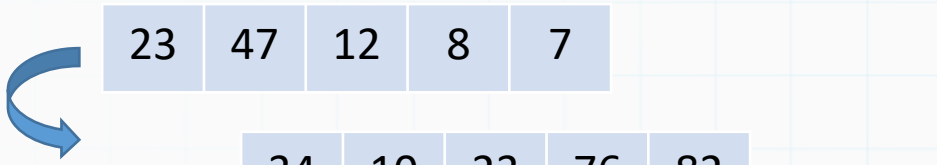
```
n?3
A
0) 5
1) 10
2) 15
B
0) 1
1) 2
2) 3
C
0) 100
1) 200
2) 300
D=[5, 1, 100, 10, 2, 200, 15, 3, 300]
```

Dica: Comece o vetor D vazio e insere (append) alternadamente os elementos de A,B e C

Listas - LAB



Exercício 4): Fazer o Exercício 2) mas agora só podemos usar dois níveis de laço (isto é, 2 níveis de laço um dentro do outro, não quer dizer apenas duas estruturas de laços, pode usar quantas estruturas de laço quiser, mas no máximo em dois níveis)

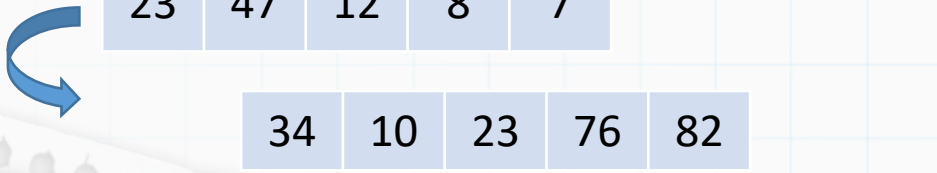


23	47	12	8	7

34	10	23	76	82

25	11	32	23	99

3 níveis

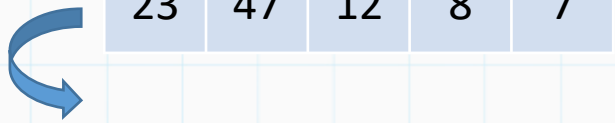


23	47	12	8	7

34	10	23	76	82

2 níveis

Dica:



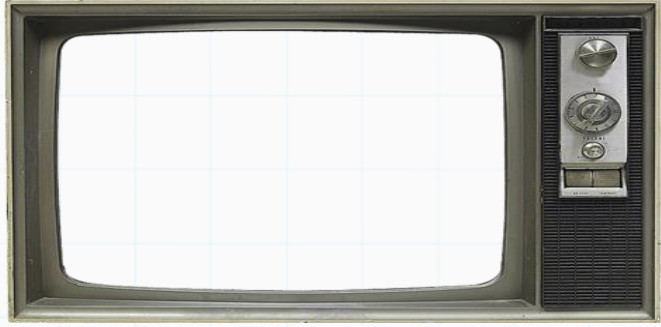
23	47	12	8	7

34	10	23	76	82

25	11	32	23	99

código

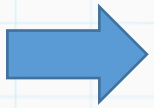
Listas - LAB



Exercício 5): Fazer o exercício 3) mas sem o comando append ou o operador de concatenação +. isto é, os vetores são criados logo no inicio com tamanhos N (A,B e C) e 3N (D)

Dica: Uma maneira de se fazer é tentar pensar numa forma de fazer um laço (while) com 2 variáveis de controle, uma para acessar o vetor D e outra para acessar os vetores A,B e C. As duas variáveis serão incrementadas em ritmos diferentes (passo)

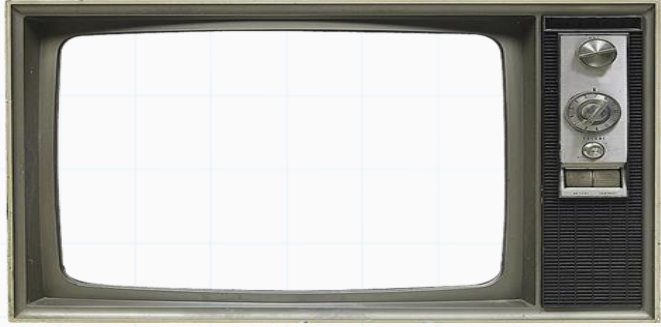
A	B	C
a_0	b_0	c_0
a_1	b_1	c_1
a_2	b_2	c_2
...	...	...
a_{N-1}	b_{N-1}	c_{N-1}

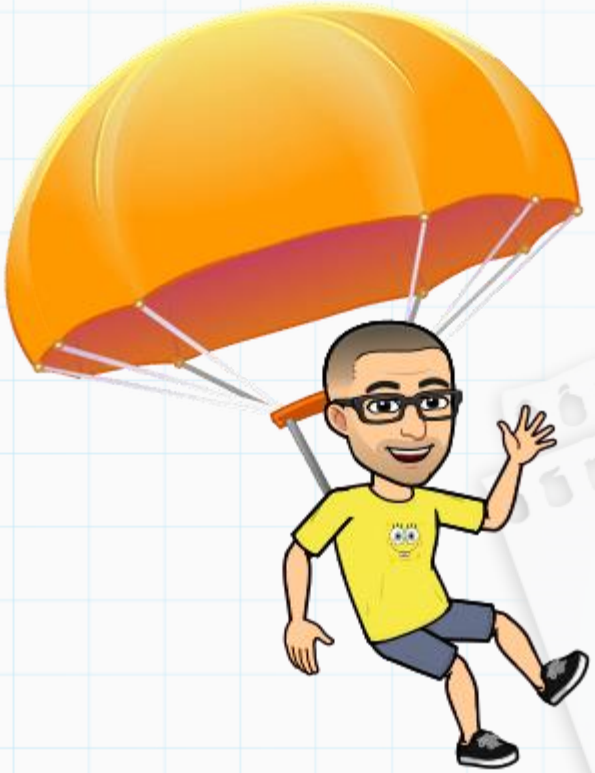


D
a_0
b_0
c_0
a_1
b_1
c_1
a_2
b_2
c_2
...

código

Até a próxima





Slides baseados no curso de Vanessa Braganholo