

Disciplina	Turma	Descrição deste documento	Prazo de Entrega
IAL002 Algoritmos e Lógica de Programação	Noite	Tarefas Preparatórias para a Aula 11	Não há entrega

### Aula de Exercícios – Problemas usando Listas

Faça todos os programas listados abaixo usando listas. Considerando que o objetivo desta disciplina é o estudo dos algoritmos, não use funções prontas das listas. Faça seus próprios algoritmos. (Isso não se aplica ao append ou insert, sem os quais fica complicado inserir novos elementos nas listas).

1. Escreva um programa que leia do teclado uma lista com tamanho de 10 elementos e exiba-o na tela um elemento por linha.
2. Escreva um programa que leia do teclado uma lista com tamanho de 10 elementos e exiba-o na tela na ordem inversa à ordem de leitura, sendo um elemento por linha da tela.
3. Escreva um programa que leia um número N (obrigatoriamente entre 0 e 50) e em seguida leia N números reais em uma lista A. Exiba a lista na tela, um elemento por linha.
4. Escreva um programa que leia um número N (obrigatoriamente entre 0 e 50) e em seguida preencha uma lista com N elementos inteiros gerados aleatoriamente entre 0 e 1000 (usar a função randint da biblioteca random – veja o quadro sobre isso no exercício 9).
5. Escreva um programa que leia um número N (obrigatoriamente entre 0 e 50 – o programa deve garantir isso em um laço) e em seguida leia N números reais em uma lista A. O programa deve separar os valores lidos em A em outras duas listas NEG e POS, a primeira contendo somente os valores negativos e a segunda contendo os valores positivos e zero. Apresentar na tela as listas NEG e POS e a quantidade de valores contidos em cada uma.
6. Leia dois números inteiros LMin e LMax. Em seguida leia 10 valores inteiros e inserindo-os em uma lista A somente se o valor fornecido estiver no intervalo [LMin, LMax]. Valores fora deste intervalo devem ser ignorados. Ao final, apresentar a lista A e seu tamanho efetivo. Observe que para este programa funcionar apropriadamente é necessário que LMin seja menor que LMax. Caso o usuário digite LMax menor que LMin, o programa deve inverter os valores.
7. Refaça o exercício anterior generalizando-o para N valores inteiros digitados no teclado, onde N é um número fornecido pelo usuário. Esse N deve ser usado no programa ao invés do tamanho fixo de 10 sugerido no programa anterior.
8. Refaça o exercício anterior de modo que os valores inválidos, ou seja, os que estão fora do intervalo [Min, Max] sejam inseridos em uma segunda lista chamada R. Apresentar na tela a lista de valores aceitos (lista A) e a lista de valores rejeitados (lista R), bem como o tamanho de cada um.
9. Escreva um programa que leia um número inteiro N e preencha uma lista com N elementos inteiros gerados aleatoriamente entre 0 e 1000 (usar a função randint da biblioteca random). Exiba na tela a lista gerada. Em seguida, o programa deve ler um valor X e informar se tal valor está, ou não está presente na lista. Neste exercício é permitido usar os operadores **in** e/ou **not in**.

Dica: como usar randint

```
from random import randint
```

```
z = randint(50, 250)
```

## Para usar a função randint é necessário importá-la para o programa  
## colocando esta linha no início do mesmo

## Este comando gera um número inteiro entre 50 e 250 e o coloca em z.

10. Escreva um programa que leia um número inteiro N e preencha uma lista com N elementos inteiros gerados aleatoriamente entre 0 e 1000 (usar a função `randint` da biblioteca `random`). Exiba na tela a lista gerada. Em seguida, o programa deve ler um valor X e informar se tal valor está, ou não está presente, bem como informar a posição de X na lista. Se houver mais de uma ocorrência de X na lista informe todas as posições. Neste exercício não é permitido usar os operadores `in` e `not in`. Também não é permitido usar qualquer função pronta de Python.
11. Escreva um programa que leia um número inteiro N e preencha uma lista com N elementos inteiros gerados aleatoriamente entre 0 e 1000. Exiba na tela a lista gerada. Em seguida, o programa deve ler um valor X e, caso X esteja na lista, deve eliminá-lo. Caso haja várias ocorrências de X, todas devem ser eliminadas. Pesquise como usar a função **`del`** (você vai precisar dela e neste exercício será permitido usá-la).
12. Escreva um programa que leia um número inteiro N e preencha uma lista com N elementos inteiros digitados pelo usuário. Exiba na tela a lista preenchida. Em seguida, o programa deve procurar e eliminar os elementos que eventualmente estiverem repetidos deixando apenas a primeira ocorrência de cada valor. Apresentar a lista resultante na tela.
13. Escreva um programa que leia um número inteiro N. Em seguida calcule e armazene em uma lista os N primeiros números primos. Exibir a lista no final. Ex. se N fornecido pelo usuário for 10, então a lista será carregada com:

V: 

2	3	5	7	11	13	17	19	23	29
---	---	---	---	----	----	----	----	----	----

14. Escreva um programa que leia do teclado duas listas com tamanho 10, com números inteiros. Em seguida o programa deve juntar as duas listas em uma única lista com o tamanho 20.
15. Escreva um programa que preencha com números inteiros duas listas denominadas A e B com diferentes tamanhos  $n_A$  e  $n_B$ , respectivamente. Em seguida o programa deve juntar as duas listas em uma única lista com o tamanho  $n_A + n_B$ . Exibir na tela a lista resultante. Veja o exemplo:

$n_A = 5$	16	8	25	12	19	}	$n_R = 12$	16	8	25	12	19	5	14	3	27	8	21	44	
$n_B = 7$							R													
A																				
B	5	14	3	27	8		21	44												

16. Escreva um programa que leia uma lista com N números inteiros, onde N é um número inteiro previamente digitado pelo usuário. O programa não deve aceitar um número digitado que já esteja inserido na lista, sendo que quando esta situação ocorrer uma mensagem deve ser dada ao usuário. Por fim exibir na tela a lista resultante.
17. Escreva um programa que leia do teclado dois números inteiros  $n_A$  e  $n_B$  e leia também duas listas denominados A e B com os tamanhos  $n_A$  e  $n_B$ , respectivamente. Na leitura de cada uma das listas é obrigatório que não sejam aceitos valores repetidos. Em seguida, o programa deve juntar as duas listas em um única lista R (resultante) tomando o cuidado de que a lista R não deve ter valores duplicados. Veja o exemplo:

$n_A = 5$	16	8	25	12	19	}	$n_R = 11$	16	8	25	12	19	5	14	3	27	21	44	
$n_B = 7$							R												
A																			
B	5	14	3	27	8		21	44											

(note que o valor 8 da lista B não foi incluído na lista resultante)

18. O programa deverá ler dois inteiros chamados Min e Max. Min pode ser qualquer valor e Max, obrigatoriamente, deve ser maior que Min. Em seguida preencher a lista com todos os valores situados entre Min e Max que sejam divisíveis por 7. Exibir a lista resultante na tela.
19. Faça um programa que permaneça em laço até que seja digitado um valor menor ou igual a zero. Cada valor válido (positivo) digitado deve ser inserido em uma lista na posição imediatamente antes do primeiro elemento que seja maior que valor que está sendo inserido. Isso resultará em uma lista ordenada de forma crescente. Será necessário usar o método **insert** da lista – pesquise sobre ele.
20. Faça um programa que leia um número inteiro N obrigatoriamente maior que 10. Preencha uma lista de tamanho N com números inteiros aleatórios. Exiba na tela a lista gerada e em seguida coloque-a em ordem crescente usando o método da bolha. Não é permitido usar o método **sort** da lista