



Gabarito da Lista de Exercícios

1.

```
# lê o custo de fábrica
custo_fabrica = float(input("Digite o custo de fábrica do carro: R$ "))

# calcula a porcentagem do distribuidor e dos impostos (sobre o custo
# de fábrica)
percent_distribuidor = 0.12 * custo_fabrica
impostos = 0.30 * custo_fabrica

# calcula o custo final ao consumidor
custo_final = custo_fabrica + percent_distribuidor + impostos

# exibe o resultado
print(f"Custo ao consumidor: R$ {custo_final:.2f}")
```

2.

```
# lê os valores de A e B
A = int(input("Digite o valor de A: "))
B = int(input("Digite o valor de B: "))

# faz a troca dos valores
temp = A
A = B
B = temp

# imprime os valores trocados
print(f"Depois da troca: A = {A}, B = {B}")
```

3.

```
# lê o tamanho do arquivo em MB
tamanho_arquivo = float(input("Digite o tamanho do arquivo para download (MB): "))

# lê a velocidade do link em Mbps
velocidade_link = float(input("Digite a velocidade do link de Internet (Mbps): "))

# converte a velocidade para MBps (Megabytes por segundo)
# 1 byte = 8 bits, então MBps = Mbps / 8
velocidade_MBps = velocidade_link / 8

# calcula o tempo em segundos
tempo_segundos = tamanho_arquivo / velocidade_MBps

# converte para minutos
tempo_minutos = tempo_segundos / 60

# imprime o resultado
print(f"Tempo aproximado de download: {tempo_minutos:.2f} minutos")
```

4.

```
# leitura das notas
P1 = float(input("Digite a nota da prova 1 (P1): "))
P2 = float(input("Digite a nota da prova 2 (P2): "))
T1 = float(input("Digite a nota do trabalho 1 (T1): "))
T2 = float(input("Digite a nota do trabalho 2 (T2): "))

# cálculo das médias
MP = (P1 + P2) / 2
MT = (T1 + T2) / 2
MF = 0.8 * MP + 0.2 * MT

# verificação da situação
if MF >= 6.0:
    print("Aprovado")
else:
    print("Não aprovado")
```

5.

```
# lê a temperatura em graus Celsius
temperatura = float(input("Digite a temperatura em °C: "))
```

```
# classificação da temperatura
if temperatura < 0:
    print("Frio extremo")
elif 0 <= temperatura <= 10:
    print("Frio")
elif 11 <= temperatura <= 25:
    print("Ameno")
elif 26 <= temperatura <= 35:
    print("Quente")
else:
    print("Muito quente")
```

6.

```
print("Menu de conversão:")
print("1. Dólar")
print("2. Euro")
print("3. Libra")
print("4. Iene")
```

```
opcao = int(input("Escolha a moeda para conversão (1-4): "))
```

```
if opcao == 1:
    taxa = 0.19
    moeda = "Dólar"
elif opcao == 2:
    taxa = 0.17
    moeda = "Euro"
elif opcao == 3:
    taxa = 0.15
    moeda = "Libra"
elif opcao == 4:
    taxa = 25
    moeda = "Iene"
else:
    taxa = None
```

```
if taxa is not None:
    valor_reais = float(input("Digite o valor em reais (R$): "))
    valor_convertido = valor_reais * taxa
    print(f"{valor_reais:.2f} reais equivalem a {valor_convertido:.2f} {moeda}")
else:
```

```
print("Opção inválida.")
```

7.

```
saldo = 1000.00
```

```
while True:
```

```
    print("\nMenu:")
```

```
    print("1. Depositar")
```

```
    print("2. Sacar")
```

```
    print("3. Consultar Saldo")
```

```
    print("4. Sair")
```

```
opcao = input("Escolha uma opção (1-4): ")
```

```
if opcao == '1':
```

```
    valor = float(input("Digite o valor a depositar: R$ "))
```

```
    if valor > 0:
```

```
        saldo += valor
```

```
        print(f"Depósito realizado. Novo saldo: R$ {saldo:.2f}")
```

```
    else:
```

```
        print("Valor inválido. O depósito deve ser maior que zero.")
```

```
elif opcao == '2':
```

```
    valor = float(input("Digite o valor a sacar: R$ "))
```

```
    if valor <= saldo and valor > 0:
```

```
        saldo -= valor
```

```
        print(f"Saque realizado. Novo saldo: R$ {saldo:.2f}")
```

```
    elif valor <= 0:
```

```
        print("Valor inválido. O saque deve ser maior que zero.")
```

```
    else:
```

```
        print("Saldo insuficiente para saque.")
```

```
elif opcao == '3':
```

```
    print(f"Saldo atual: R$ {saldo:.2f}")
```

```
elif opcao == '4':
```

```
    print("Encerrando o programa. Obrigado!")
```

```
    break
```

```
else:
```

```
    print("Opção inválida. Digite um número entre 1 e 4.")
```

8.

```

def potencia(base, expoente):
    if expoente == 0:
        return 1
    resultado = 1
    for _ in range(expoente):
        resultado *= base
    return resultado

# leitura dos valores
x = int(input("Digite o valor de x (base): "))
y = int(input("Digite o valor de y (expoente): "))

# cálculo da potência
resultado = potencia(x, y)

# exibe o resultado
print(f"{x}^{y} = {resultado}")

```

9.

```

def imprimir_multiplos(x):
    print(f"Múltiplos de {x} entre 1 e 100:")
    for num in range(1, 101):
        if num % x == 0:
            print(num, end=' ')
    print() # para pular linha no final

# entrada do usuário
x = int(input("Digite um número inteiro: "))

imprimir_multiplos(x)

```

10.

```

# Lista que armazena as tarefas
tarefas = []

def adicionar_tarefa():
    tarefa = input("Digite a nova tarefa: ")

    if tarefa:
        tarefas.append(tarefa)

```

```
print("Tarefa adicionada com sucesso!")

else:
    print("Tarefa não pode ser vazia.")


def listar_tarefas():
    if tarefas:
        print("\nLista de Tarefas:")
        indice = 1
        for tarefa in tarefas:
            print(f"{indice}. {tarefa}")
            indice += 1
    else:
        print("Nenhuma tarefa cadastrada.")


def remover_tarefa():
    nome = input("Digite o nome exato da tarefa que deseja remover: ")
    if nome in tarefas:
        tarefas.remove(nome)
        print("Tarefa removida com sucesso!")
    else:
        print("Tarefa não encontrada.")


def menu():
    while True:
        print("\nMenu de Tarefas")
        print("1. Adicionar uma nova tarefa")
        print("2. Listar todas as tarefas")
```

```
print("3. Remover uma tarefa pelo nome")
print("4. Sair")
```

```
opcao = input("Escolha uma opção (1-4): ")
```

```
if opcao == '1':
    adicionar_tarefa()
elif opcao == '2':
    listar_tarefas()
elif opcao == '3':
    remover_tarefa()
elif opcao == '4':
    print("Encerrando o programa. Até mais!")
    break
else:
    print("Opção inválida. Tente novamente.")
```

```
# Executa o menu
```

```
menu()
```

11.

```
# Lista de ocupação: False = livre, True = ocupado
assentos = [False] * 10
```

```
def reservar_assento():
```

```
    numero = int(input("Digite o número do assento para reservar (1 a
10): "))
```

```
    if 1 <= numero <= 10:
```

```
if assentos[numero - 1] == False:  
    assentos[numero - 1] = True  
    print(f"Assento {numero} reservado com sucesso.")  
else:  
    print(f"O assento {numero} já está ocupado.")  
  
else:  
    print("Número de assento inválido.")  
  
  
def liberar_assento():  
    numero = int(input("Digite o número do assento para liberar (1 a  
    10): "))  
    if 1 <= numero <= 10:  
        if assentos[numero - 1] == True:  
            assentos[numero - 1] = False  
            print(f"Assento {numero} liberado com sucesso.")  
        else:  
            print(f"O assento {numero} já está livre.")  
    else:  
        print("Número de assento inválido.")  
  
  
def mostrar_mapa():  
    print("\nMapa de Ocupação dos Assentos:")  
    for i in range(10):  
        if assentos[i] == True:  
            status = "Ocupado"  
        else:  
            status = "Livre"
```

```
print(f"Assento {i+1}: {status}")

def menu():
    while True:
        print("\n--- Menu do Cinema ---")
        print("1. Reservar um assento")
        print("2. Liberar um assento")
        print("3. Mostrar mapa de ocupação")
        print("4. Sair")

        opcao = input("Escolha uma opção (1-4): ")

        if opcao == '1':
            reservar_assento()
        elif opcao == '2':
            liberar_assento()
        elif opcao == '3':
            mostrar_mapa()
        elif opcao == '4':
            print("Encerrando o programa. Até logo!")
            break
        else:
            print("Opção inválida. Tente novamente.")

def main():
    menu()
```

```
# Ponto de entrada

if __name__ == "__main__":
    main()

12.

# Lista de tuplas no formato (nome, nota, disciplina)
notas = []

def adicionar_nota():
    nome = input("Nome do aluno: ")
    nota = float(input("Nota do aluno: ")) # sem verificação
    disciplina = input("Disciplina: ")
    notas.append((nome, nota, disciplina))
    print("Nota adicionada com sucesso.")

def melhor_por_disciplina():
    if len(notas) == 0:
        print("Nenhuma nota cadastrada.")
        return

    disciplinas = []
    for n in notas:
        if n[2] not in disciplinas:
            disciplinas.append(n[2])

    print("\nMelhor aluno por disciplina:")
    for d in disciplinas:
```

```
melhor_nota = -1
melhor_aluno = ""

for n in notas:
    if n[2] == d and n[1] > melhor_nota:
        melhor_nota = n[1]
        melhor_aluno = n[0]

print(f"{d}: {melhor_aluno} ({melhor_nota})")

def consultar_por_aluno():
    nome_busca = input("Digite o nome do aluno: ")
    encontrou = False

    for n in notas:
        if n[0].lower() == nome_busca.lower():
            print(f"{n[2]}: {n[1]}")
            encontrou = True

    if not encontrou:
        print("Nenhuma nota encontrada para este aluno.")

# Função auxiliar para usar como chave no sorted()
def obter_nota(tupla):
    return tupla[1]

def exibir_ordenadas():
    if len(notas) == 0:
        print("Nenhuma nota cadastrada.")

    return
```

```
ordenadas = sorted(notas, key=obter_nota, reverse=True)
print("\nNotas ordenadas (decrescente):")
for n in ordenadas:
    print(f"{n[1]:.2f}, {n[0]}, {n[2]}")

def menu():
    while True:
        print("\n--- MENU DE NOTAS ---")
        print("1. Adicionar nota")
        print("2. Mostrar melhor aluno por disciplina")
        print("3. Consultar notas por aluno")
        print("4. Exibir notas ordenadas (decrescente)")
        print("5. Sair")

    opcao = input("Escolha uma opção (1-5): ")

    if opcao == '1':
        adicionar_nota()
    elif opcao == '2':
        melhor_por_disciplina()
    elif opcao == '3':
        consultar_por_aluno()
    elif opcao == '4':
        exibir_ordenadas()
    elif opcao == '5':
        print("Encerrando o programa.")
        break
```

```
    else:  
        print("Opção inválida. Tente novamente.")
```

```
def main():  
    menu()  
  
if __name__ == "__main__":  
    main()
```

13.

```
# Conjuntos para armazenar os nomes dos alunos  
palestra_ia = set()  
workshop_python = set()
```

```
def adicionar_aluno():  
    nome = input("Nome do aluno: ").strip()  
    evento = input("Evento (IA ou Python): ").strip().lower()  
    if evento == "ia":  
        palestra_ia.add(nome)  
        print(f"{nome} adicionado à palestra de IA.")  
    elif evento == "python":  
        workshop_python.add(nome)  
        print(f"{nome} adicionado ao workshop de Python.")  
    else:  
        print("Evento inválido. Digite apenas 'IA' ou 'Python'.")
```

```
def mostrar_ambos():  
    ambos = palestra_ia.intersection(workshop_python)
```

```
print("\nAlunos que participaram de ambos os eventos:")
if ambos:
    for aluno in ambos:
        print(aluno)
else:
    print("Nenhum aluno participou de ambos os eventos.")

def mostrar_so_ia():
    somente_ia = palestra_ia.difference(workshop_python)
    print("\nAlunos que participaram somente da palestra de IA:")
    if somente_ia:
        for aluno in somente_ia:
            print(aluno)
    else:
        print("Nenhum aluno participou somente da palestra de IA.")

def mostrar_pelo_menos_um():
    pelo_menos_um = palestra_ia.union(workshop_python)
    print("\nAlunos que participaram de pelo menos um evento:")
    if pelo_menos_um:
        for aluno in pelo_menos_um:
            print(aluno)
    else:
        print("Nenhum aluno participou de eventos.")

def verificar_participacao():
    nome = input("Digite o nome do aluno a verificar: ").strip()
```

```
em_ia = nome in palestra_ia
em_python = nome in workshop_python

print()
if em_ia and em_python:
    print(f"{nome} participou de ambos os eventos.")
elif em_ia:
    print(f"{nome} participou somente da palestra de IA.")
elif em_python:
    print(f"{nome} participou somente do workshop de Python.")
else:
    print(f"{nome} não participou de nenhum evento.")

def menu():
    while True:
        print("\n--- MENU ---")
        print("1. Adicionar aluno a um evento")
        print("2. Mostrar alunos que participaram de ambos os eventos")
        print("3. Mostrar alunos que participaram somente da palestra de IA")
        print("4. Mostrar alunos que participaram de pelo menos um evento")
        print("5. Verificar participação de um aluno")
        print("6. Sair")

    opcao = input("Escolha uma opção: ").strip()

    if opcao == "1":
```

```
    adicionar_aluno()

    elif opcao == "2":
        mostrar_ambos()

    elif opcao == "3":
        mostrar_so_ia()

    elif opcao == "4":
        mostrar_pelo_menos_um()

    elif opcao == "5":
        verificar_participacao()

    elif opcao == "6":
        print("Encerrando o programa.")

        break

    else:
        print("Opção inválida. Tente novamente.")

def main():

    menu()

if __name__ == "__main__":
    main()
```

14.

```
estoque = {}

def adicionar_livro():

    titulo = input("Título do livro: ").strip()

    qtd = int(input("Quantidade a adicionar: "))
```

```
if titulo in estoque:  
    estoque[titulo] += qtd  
  
else:  
    estoque[titulo] = qtd  
  
print(f'{qtd} unidade(s) de '{titulo}' adicionada(s).')
```

```
def remover_livro():  
    titulo = input("Título do livro: ").strip()  
  
    if titulo not in estoque:  
        print("Livro não encontrado no estoque.")  
  
    return
```

```
qtd = int(input("Quantidade a remover: "))
```

```
if qtd > estoque[titulo]:  
    print(f'Não há unidades suficientes para remover. Estoque atual:  
{estoque[titulo]}')  
  
    return
```

```
estoque[titulo] -= qtd  
  
print(f'{qtd} unidade(s) removida(s) de '{titulo}'.")  
  
if estoque[titulo] == 0:  
    print(f'Estoque do livro '{titulo}' zerado.')
```

```
def consultar_livro():  
    titulo = input("Título do livro: ").strip()  
  
    if titulo in estoque:
```

```
print(f"Quantidade disponível de '{titulo}': {estoque[titulo]}")  
else:  
    print("Livro não está no estoque.")  
  
def mostrar_estoque():  
    if not estoque:  
        print("Estoque vazio.")  
        return  
  
    print("\nLivros em estoque:")  
    for titulo in sorted(estoque.keys()):  
        print(f"{titulo}: {estoque[titulo]}")  
  
def menu():  
    while True:  
        print("\n--- MENU ---")  
        print("1. Adicionar livro ao estoque")  
        print("2. Remover unidades de um livro")  
        print("3. Consultar quantidade de um livro")  
        print("4. Mostrar todos os livros com quantidades")  
        print("5. Sair")  
  
        opcao = input("Escolha uma opção: ").strip()  
        if opcao == '1':  
            adicionar_livro()  
        elif opcao == '2':  
            remover_livro()
```

```
elif opcao == '3':
    consultar_livro()
elif opcao == '4':
    mostrar_estoque()
elif opcao == '5':
    print("Encerrando o programa.")
    break
else:
    print("Opção inválida, tente novamente.")
```

```
def main():
    menu()

if __name__ == "__main__":
    main()
```

15.

```
campeonato = {}

def adicionar_time():
    time = input("Nome do time a adicionar: ").strip()
    if time in campeonato:
        print("Este time já está cadastrado.")
    else:
        campeonato[time] = 0
        print(f"Time '{time}' adicionado com 0 pontos.")
```

```
def registrar_resultado():
```

```
time1 = input("Nome do primeiro time: ").strip()
time2 = input("Nome do segundo time: ").strip()

if time1 not in campeonato or time2 not in campeonato:
    print("Ambos os times devem estar cadastrados.")
    return

gols1 = int(input(f"Gols do {time1}: "))
gols2 = int(input(f"Gols do {time2}: "))

if gols1 > gols2:
    campeonato[time1] += 3
    print(f"{time1} venceu e ganhou 3 pontos.")

elif gols2 > gols1:
    campeonato[time2] += 3
    print(f"{time2} venceu e ganhou 3 pontos.")

else:
    campeonato[time1] += 1
    campeonato[time2] += 1
    print("Empate. Cada time ganhou 1 ponto.")

def obter_pontos(item):
    return item[1]

def mostrar_classificacao():
    if not campeonato:
        print("Nenhum time cadastrado.")
```

```
return

print("\nClassificação:")

ordenado = sorted(campeonato.items(), key=obter_pontos,
reverse=True)

for time, pontos in ordenado:
    print(f"{time}: {pontos} ponto(s)")

def remover_time():

    time = input("Nome do time a remover: ").strip()

    if time in campeonato:
        del campeonato[time]
        print(f"Time '{time}' removido do campeonato.")

    else:
        print("Time não encontrado.")

def menu():

    while True:

        print("\n--- MENU DO CAMPEONATO ---")

        print("1. Adicionar time")
        print("2. Registrar resultado de partida")
        print("3. Mostrar classificação")
        print("4. Remover time")
        print("5. Sair")

        opcao = input("Escolha uma opção: ").strip()
```

```
if opcao == '1':
    adicionar_time()
elif opcao == '2':
    registrar_resultado()
elif opcao == '3':
    mostrar_classificacao()
elif opcao == '4':
    remover_time()
elif opcao == '5':
    print("Encerrando o programa.")
    break
else:
    print("Opção inválida. Tente novamente.")
```

```
def main():
    menu()

if __name__ == "__main__":
    main()
```