

Lista de Exercícios

1. Elabore um programa que permita ao usuário digitar um dicionário cujas chaves representam o código de um projeto e os valores correspondem aos respectivos tempos de execução. Em seguida, o programa deve apresentar os dados em ordem crescente do código do projeto.
2. Escreva um programa que permita ao usuário informar as coordenadas de disparos e calcular a pontuação obtida. Inicialmente o usuário informa as coordenadas x e y , uma a uma, e o cálculo da pontuação é dado por: $\text{Pontos} = 10 - d/10$ (a cada 10 unidades de distância, perde um ponto), sendo d a distância do ponto (x, y) até o centro do alvo, localizado em $(0, 0)$. Os dados devem ser armazenados em um dicionário, sendo a chave formada pela **tupla** com as coordenadas (x, y) e a pontuação obtida como o respectivo valor.
Após o cadastro, o programa deve percorrer o dicionário e exibir as coordenadas dos disparos que registraram pontuação máxima para o jogador (que não necessariamente deve ser 10).
3. (6.18) Escreva um programa que gere um dicionário, onde cada chave seja um caractere e seu valor seja o número desse caractere encontrado em uma frase lida. Desconsiderar o espaço.
Exemplo: “A cara rajada da jararaca”
 $\{ 'A': 1, 'c': 2, 'a': 10, 'r': 4, 'j': 2, 'd': 2 \}$
4. Altere o exercício anterior de tal maneira que seja gerado um dicionário com o número de vezes que uma palavra aparece em uma frase. Não devem ser considerados os espaços nem a diferença entre a caixa das letras.
5. Para ajudar a controlar gastos, elabore um programa que permita ao usuário informar as operações que estão acontecendo em sua conta bancária. O programa deverá solicitar ao usuário o tipo de operação (D)epósito, (S)aque, (P)agamento de contas e o valor da transação. Crie um dicionário que acumula os valores dos depósitos, saques e contas pagas.
Caso a operação seja (F)im, o programa deverá informar o valor restante na conta bancária.
6. Elabore um programa que permita ao usuário armazenar a placa de um veículo, bem como o número da vaga que foi sorteada durante a reunião do condomínio.
Verifique se a vaga ainda não foi preenchida ou se o veículo não está ocupando outra vaga.
Dica: utilize dois dicionários, um placa-vaga e outro vaga-placa para fazer as verificações.
Após inserir todos os veículos, exiba uma tabela com a vaga e a placa dos veículos cadastrados. A tabela deve estar ordenada pelo número da vaga.
7. Algumas atividades comuns dos PAEs possuem valores distintos de horas que os alunos podem receber. As atividades ATC003, ATC006, ATC009 e ATC030 valem 5 horas; as atividades ATC001, ATC004, ATC007, ATC010, ATC013, ATC026, ATC028, ATC032 e ATC034 valem 10 horas e as atividades ATC016, ATC018, ATC020, ATC022, ATC024 e ATC036 valem 40 horas. Crie um dicionário com o código da atividade como chave relacionado aos respectivos valores em horas.

Em seu programa permita ao usuário informar o número de PAEs que se inscreveu ao longo do ano e se recebeu o conceito “Cumpriu” ou “Não Cumpriu”. Para tal, crie um segundo dicionário relacionado o código da atividade com o conceito recebido. Caso o usuário digite um código inválido, o programa deverá avisá-lo e solicitar novamente a informação (use um **while** ao invés de um **for**).

Após digitar os códigos e conceitos, o programa deverá apresentar ao usuário o total de horas cumpridas, bem como o total de horas requisitadas. Veja o exemplo:

```
Digite o número de atividades: 4
Digite o código da atividade: ATC034
Digite o conceito recebido (C/N): C
Digite o código da atividade: ATC016
Digite o conceito recebido (C/N): C
Digite o código da atividade: ATC011
Atividade não cadastrada
Digite o código da atividade: ATC010
Digite o conceito recebido (C/N): N
Digite o código da atividade: ATC009
Digite o conceito recebido (C/N): C

Cumpriu 50 de 60 horas
```

8. Deseja-se implementar um sistema de gerenciamento de um estacionamento utilizando dicionários. Considere que a chave representa a placa do veículo e o valor o horário de entrada (convertido para minutos a partir da meia noite).

O programa deve funcionar continuamente, até que o usuário informe que não existem mais carros para entrar ou sair no dia. Utilize um **while**. Dentro desse laço, o usuário deve informar a placa do veículo e uma letra, indicando a (E)ntrada ou (S)áida do veículo. Nessa fase de testes do desenvolvimento o usuário irá digitar dois valores inteiros: a hora e os minutos.

Caso o usuário indique que o veículo está entrando no estacionamento, inclua o valor do horário (convertido para minutos a partir da meia noite) à chave dada pela placa.

Caso o usuário indique que o veículo está saindo do estacionamento, deve-se calcular o tempo de permanência (o usuário deve informar a hora e minuto de saída; convertendo o valor para minutos a partir da meia noite) e **excluir** a chave com a placa do dicionário. Recomenda-se fazer uma verificação de pertinência antes dessas operações.

No estacionamento, o cliente paga por hora cheia. Assim, se o veículo permanecer estacionado por 2 horas e 4 minutos, o cliente deverá pagar por 3 horas. Veja o cálculo do período a ser cobrado:

$$\text{permanência} = \text{Saída}_{\text{minutos desde 0h}} - \text{Entrada}_{\text{minutos desde 0h}}$$

$$\text{permanência}_{\text{EmHoras}} = \text{Quociente de permanência por 60}$$

$$\text{permanência}_{\text{EmMinutos}} = \text{permanência} - (60 \cdot \text{permanência}_{\text{EmHoras}})$$

$$\text{HorasCobradas} = \begin{cases} \text{permanência}_{\text{EmHoras}}, & \text{se } \text{permanência}_{\text{EmMinutos}} = 0 \\ \text{permanência}_{\text{EmHoras}} + 1, & \text{caso contrário} \end{cases}$$