Monta Palavras

Sobre o problema

Considere um jogo de formar palavras. Neste jogo, cada jogador recebe um conjunto de letras e deve decidir qual palavra formada com aquelas letras vai contabilizar a maior quantidade de pontos.

Pense nas letras que são disponibilizadas para cada jogador como "pecinhas" de um jogo, ou seja, pode haver letras repetidas. Além disso, cada letra possui um valor, que ajuda a contabilizar mais pontos na palavra que o jogador formar.

Para formar uma palavra, todas as letras que a compõem devem estar presentes no conjunto de entrada. Em contrapartida, nem todas as letras disponíveis precisam ser usadas. Por exemplo, se você possui as letras "`ybttaaa`", você pode formar a palavra "`batata`", deixando de fora a letra "`y`".

Passos para a execução

Você pode usar a linguagem que preferir, mas seu programa deverá ser **executado via terminal**. Sua tarefa é implementar um programa que lide com uma jogada do jogo de Montar Palavras, sempre fazendo **a melhor jogada possível**.

Para isso você deve:

- Disponibilizar um campo de entrada no qual será informado quais letras estão disponíveis para que a palavra seja formada.
- Mostrar qual é a palavra de maior pontuação, juntamente com qual pontuação foi obtida.
- Mostrar quais letras não foram utilizadas para formar a melhor palavra.
- Tratar quaisquer caracteres especiais como letras não usadas ("pecinhas" que sobraram).
- Escrever no cabeçalho da classe principal do seu programa (via comentário) qual a lógica utilizada no funcionamento do programa. Se alguma estrutura de dados especial tiver sido utilizada e for relevante no funcionamento do algoritmo, citá-la também.
  Exemplo:
  `/**`

```
 * A lógica do programa se baseia em percorrer (...)
 * A contagem dos pontos é feita no momento em que (...)
 */
```

Regras

- O valor de cada letra é fixo e informado abaixo.
- O banco de palavras também é fixo e informado abaixo. Considere que não existem palavras que não estejam no banco.
- O valor da palavra corresponde à soma dos valores de cada letra que a compõem. O valor das letras que não foram utilizadas para formar a palavra não é descontado no processo.
- Em caso de **empate** no valor de duas palavras, a **palavra mais curta** deverá ser escolhida. Exemplo: "`nada`" (5 pontos) e "`meu`" (também 5 pontos) => a palavra "`meu`" deverá ser escolhida
- **Se ainda assim houver empate**, a palavra que vem primeiro em uma **organização alfabética** deve ser escolhida. Exemplo: "`nada`" (5 pontos) e "`lado`" (também 5 pontos) => a palavra “lado” deverá ser escolhida.
- **Desconsiderar** acentos e diferenças entre letras maiúsculas e minúsculas.
- **Não copiar** nenhuma solução de terceiros. Esperamos que você crie sua própria solução para o problema. Você está livre para acessar a Internet a fim de solucionar dúvidas relacionadas à plataforma para a qual você está desenvolvendo, estruturas de dados que você vier a utilizar, documentação da linguagem, etc.

Valor das letras

- 1 ponto: E, A, I, O, N, R, T, L, S, U
- 2 pontos: D, G
- 3 pontos: B, C, M, P
- 5 pontos: F, H, V
- 8 pontos: J, X
- 13 pontos: Q, Z

Banco de palavras

"Abacaxi", "Manada", "mandar", "porta", "mesa", "Dado", "Mangas", "Já", "coisas", "radiografia", "matemática", "Drogas", "prédios", "implementação", "computador", "balão", "Xícara", "Tédio", "faixa", "Livro", "deixar", "superior", "Profissão", "Reunião", "Prédios", "Montanha", "Botânica", "Banheiro", "Caixas", "Xingamento", "Infestação", "Cupim", "Premiada", "empanada", "Ratos", "Ruído", "Antecedente", "Empresa", "Emissário", "Folga", "Fratura", "Goiaba", "Gratuito", "Hídrico", "Homem", "Jantar", "Jogos", "Montagem", "Manual", "Nuvem", "Neve", "Operação", "Ontem", "Pato", "Pé", "viagem", "Queijo", "Quarto", "Quintal", "Solto", "rota", "Selva", "Tatuagem", "Tigre", "Uva", "Último", "Vitupério", "Voltagem", "Zangado", "Zombaria", "Dor"

Critérios de avaliação

1. Compila e executa sem crashar
3. Funcionamento correto do programa
4. Performance da solução implementada
5. Descrição do funcionamento do programa (deve condizer com o que foi implementado)
6. Tratamento da entrada / caracteres especiais
7. Organização e clareza do código

Observações

- Deixe claro como seu programa deve ser executado por nós. Scripts de automatização são sempre bem-vindos.
- Documente sua lógica de implementação para entendermos o máximo possível do seu programa.
- Seu programa deverá continuar aceitando novas entradas e imprimindo resultados sem a necessidade de executá-lo novamente.
- Considere que as entradas serão sempre letras sem caracteres especiais, mas aceite letras maiúsculas e minúsculas. ([a-zA-Z])

Exemplos de entrada e saída

Você deve ler as letras de entrada a partir da entrada padrão e imprimir seu resultado seguindo o formato:

- Exemplo com palavra encontrada e letras sobrando:

```
# Digite as letras disponíveis nesta jogada: yibttaaa
#
# BATATA, palavra de 8 pontos
# Sobraram: I, Y
```

- Exemplo com palavra encontrada sem sobra de letras:

```
# Digite as letras disponíveis nesta jogada: bttaaa
#
# BATATA, palavra de 8 pontos
```

- Exemplo com nenhuma palavra encontrada:

```
# Digite as letras disponíveis nesta jogada: o
#
# Nenhuma palavra encontrada
# Sobraram: O
```

Pontos extras

As funcionalidades a seguir **não são obrigatórias** para que seu programa seja avaliado, mas contarão como pontos extras para avaliarmos seu nível técnico de programação. Fique à vontade para escolher um ou mais itens abaixo para usar na sua implementação.

★ **Múltiplas palavras**

Permita que seu programa responda com múltiplas palavras nos casos em que é possível gerar um score mais alto montando duas ou mais palavras ao invés de uma única.

Exemplos:

```
# Digite as letras disponíveis nesta jogada: drpatou
#
# PORTA, total de 7 pontos
# Sobraram: D, U

# Digite as letras disponíveis nesta jogada: drpatovu
#
# UVA, DOR, total de 11 pontos
# Sobraram: P, T
```

★ **Bônus de posição**

Aceite que o usuário possa definir uma posição na palavra que dobre o valor da letra que for posicionada ali.

A posição bônus deve ser um valor maior ou igual a 1, que corresponda ao espaço em uma palavra da esquerda para a direita (a posição 2 na palavra “ar” seria o espaço da letra R).

Se o usuário entrar com uma posição bônus de valor 0, o programa deve desconsiderar o bônus.

Exemplos:

```
# Digite as letras disponíveis nesta jogada: folgauv
# Digite a posição bônus: 0
#
# FOLGA, palavra de 10 pontos
# Sobraram: U, V

# Digite as letras disponíveis nesta jogada: folgauv
# Digite a posição bônus: 2
#
# UVA, palavra de 12 pontos
# Sobraram: F, O, L, G
```

Boa sorte! =)