

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



Passos para a execução

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 desafio precisa ser feito usando os frameworks nativos ou do Android ou do iOS, ou seja, usando o Android Studio (Kotlin ou Java) ou o XCode (Swift ou Objective C).
- 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 a soma dos valores de cada letra que a compõe. 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, não importa qual palavra será escolhida. Exemplo: "nada" (5 pontos) e "lado" (também 5 pontos) => não importa qual das duas será 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

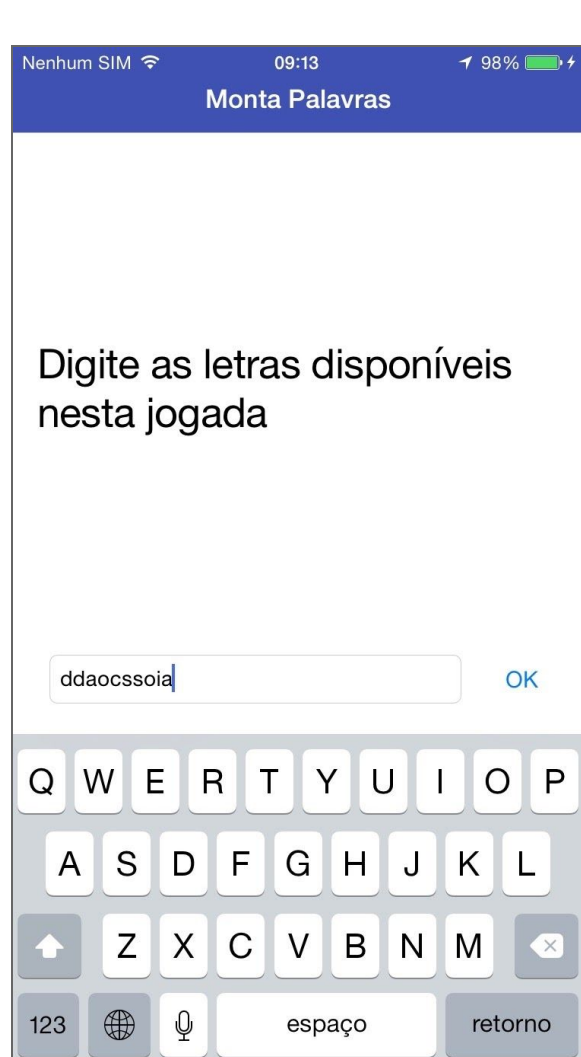
- 1 ponto: E, A, I, O, N, R, T, L, S, U
- 2 pontos: W, D, G
- 3 pontos: B, C, M, P
- 4 pontos: F, H, V
- 8 pontos: J, X
- 10 pontos: Q, Z

Banco de palavras

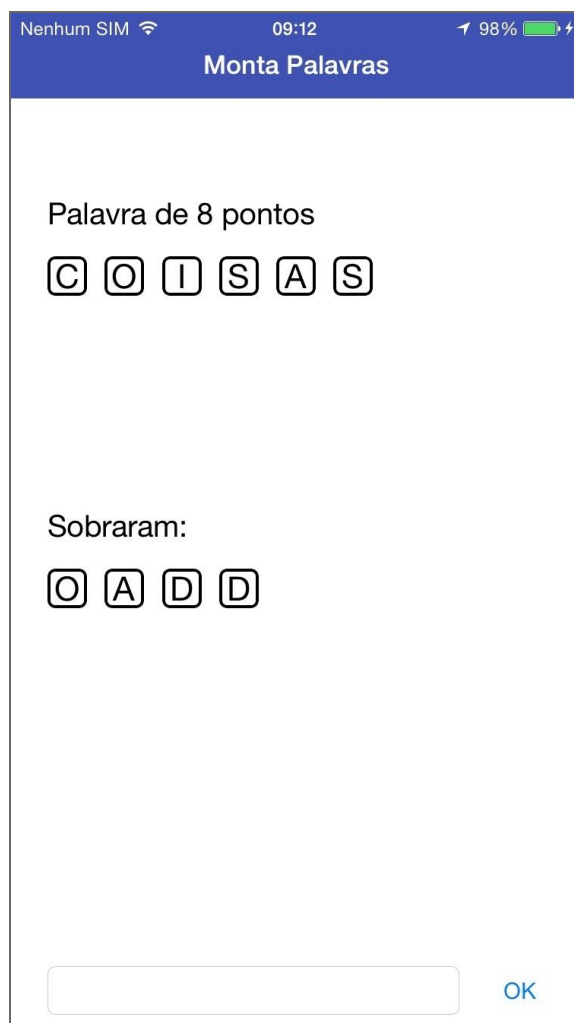
"Abacaxi", "Manada", "mandar", "porta", "mesa", "Dado", "Mangas", "Já", "coisas", "radiografia", "matemática", "Drogas", "prédios", "implementação", "computador", "balão", "Xícara", "Tédio", "faixa", "Livro", "deixar", "superior", "Profissão", "Reunião", "Prédios", "Montanha", "Botânica", "Banheiro", "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"

Exemplos

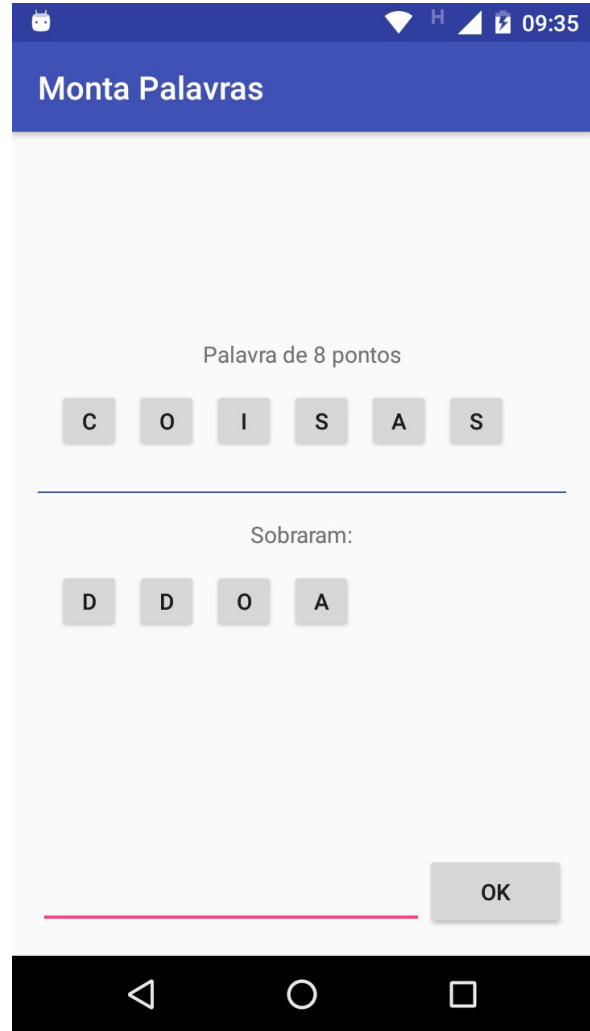
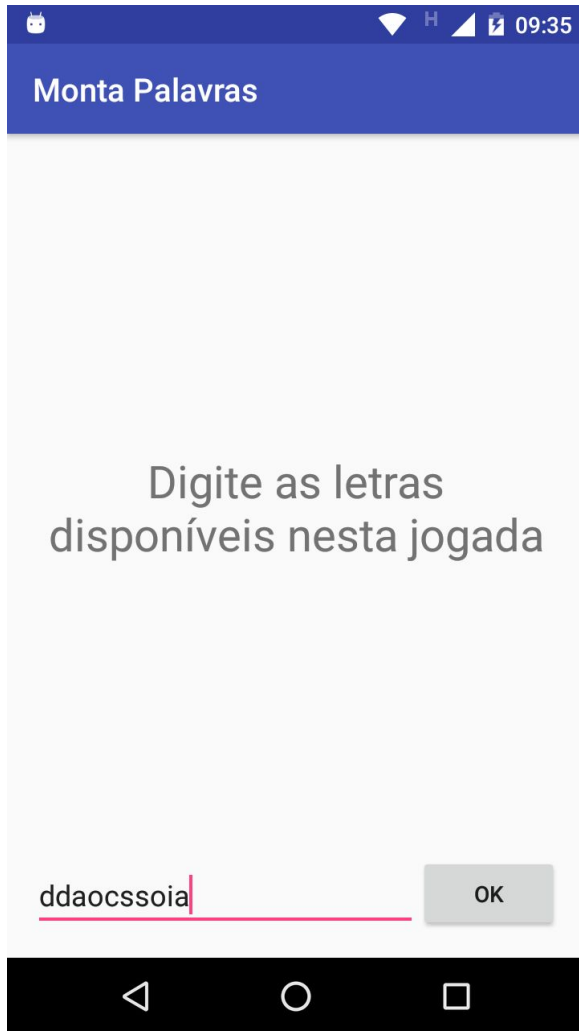
Os exemplos abaixo servem como guia. Seu layout deve ficar o mais próximo possível destes modelos.



iOS



Android



Critérios de avaliação

1. (1 pt) Compila e executa sem crashar
3. (6 pts) Funcionamento correto do programa
4. (3 pts) Performance da solução implementada
5. (1 pt) Descrição do funcionamento do programa (deve condizer com o que foi implementado)
6. (1 pt) Tratamento da entrada / caracteres especiais
7. (4 pts) Layout condizente com os exemplos
8. (4 pts) Organização e clareza do código

Implementação via terminal

Alternativamente, você pode criar sua solução sem lidar com Android ou iOS, caso você não esteja familiarizado com a programação para estas plataformas. Neste caso, você pode utilizar as linguagens Java, Swift ou C / C++. Naturalmente, a maior quantidade de pontos só poderá ser obtida por aqueles que desenvolverem a solução para Android ou iOS. Mas não se preocupe: ainda assim iremos valorizar sua solução ;)

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

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