

TESTE DE DESENVOLVIMENTO DE APP

Desenvolva as questões 1 e 2 em papel ou arquivo Word. Utilize a linguagem de sua preferência. Não é necessário saber todos os recursos e sintaxe da linguagem e do sistema de cabeça. Se você quiser utilizar alguma função ou recurso, mas não lembra como, basta criar a existência de tal recurso e explicar sua funcionalidade.

Escreva o código das perguntas abaixo. Não esqueça de comentar.

A questão 3 deve ser desenvolvido o código.

Questão 1.

Dados os valores iniciais para as variáveis abaixo, encontre o resultado das operações lógicas.

unsigned int A = 0xF0FA, B = 0xAFAF;

unsigned long C = 0x0505, E;

- E = A & B;
- E = A | B;
- E = C << 8;
- E = B & 0x5A5AU;
- E = C << 16 ? B : A;

Questão 2

Implement a method to perform basic string compression using the counts of repeated characters. For example, the string `aabccccaaa` would become `a2b1c5a3`. If the "compressed" string does not become smaller than the original string, your method should return the original string, otherwise it should return the compressed string. You can assume the string has only uppercase and lowercase letters (a-z) and the compression is case insensitive.

'aabccccaaa' => 'a2b1c5a3'

'abcde' => 'abcde'

'aaaaBBBbb' => 'a4B5'

'aAaAaA' => 'a6'

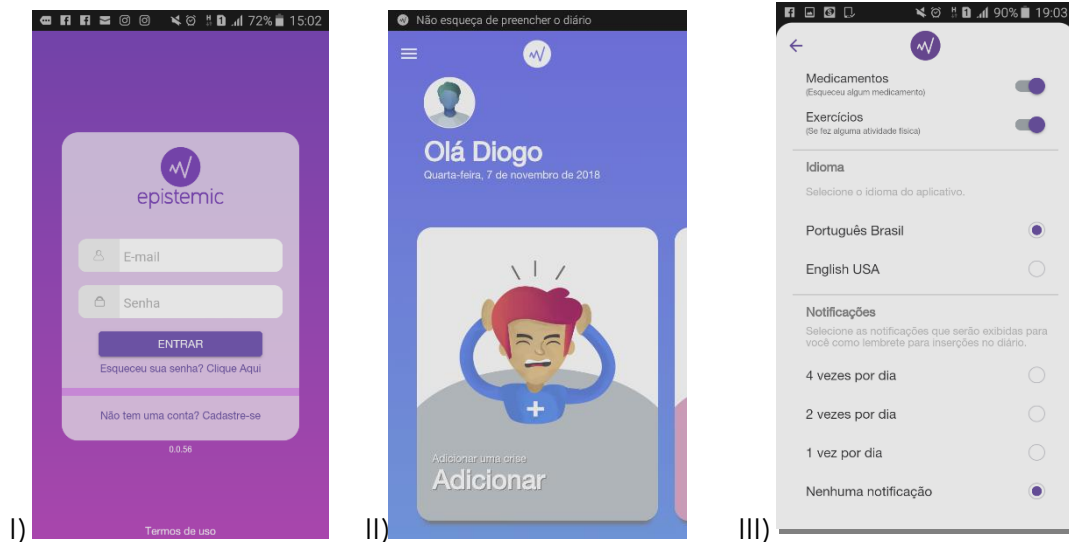
'aAbB' => 'aAbB'

'aAabAaA => 'a3b1a3'

Questão 3

Prova Prática Epistemic

Abaixo são mostradas as telas de Login, tela inicial e de configurações do app da Epistemic, que podem auxiliá-lo na prova durante esta prova.



Usando seus conhecimentos de linguagem de programação de sua preferência (Java, XML, Objective-C, Swift, Python, C, etc.), e o IDE de sua preferência (Android Studio, Xcode, etc.) realize as atividades abaixo. Lembre-se de comentar o código e destacar qual linguagem/linguagens e ferramentas que você usou.

Como output pedimos um arquivo .zip ou .rar contendo:

- um único app com seu nome e arquivo instalável (Ex: um “.apk”) contendo as funcionalidades desenvolvidas durante as atividades listadas abaixo
- o código resultante
- qualquer outro arquivo que você achar relevante (imagens, .txt, .doc, etc.)

- 1) Implemente o código da criação dos objetos de uma tela de Login contendo:
 - a. Duas caixas de texto de input para Login(e-mail) e Senha
 - b. Um botão de Entrar
 - c. Três hiperlinks para “Recuperação de Senha”, “Fazer novo Cadastro” e acessar os “Termos de Condições e Uso”
- 2) Verificação e erros:
 - a. Implemente o código referente a tela de Login para as seguintes situações abaixo:

- i. Verificar, no momento que o botão “Entrar” for pressionado, se a senha e e-mail inseridos nos respectivos campos estão corretos
 - ii. Bloquear o acesso ao usuário que tiver sua senha inserida com erro por três vezes
 - iii. Solicitar e-mail do usuário quando ele clicar em “Esqueceu sua senha”, e, caso ele já tenha inserido o e-mail na tela de Login, deixar o mesmo e-mail como default
 - b. Use, teste e demonstre as funcionalidades dos três itens em 2a com pelo menos para cada.
 - c. Implemente alertas correspondentes a situações de erros e sucesso para as três funcionalidades em 2a.
- 3) O App Epistemic notifica, com uma frequência determinada pelo usuário, a possibilidade de inserir no diário dados de eventos ocorridos no dia. Abaixo mostramos uma parte do código do aplicativo (um método), em Java, que realiza o registro da frequência com a qual as notificações serão enviadas. Essa função é ativada quando o usuário seleciona quantas notificações ele deseja receber por dia na tela de configurações. Leia o código disponibilizado e realize as seguintes tarefas:
- a. Faça um (ou mais) fluxograma(s) explicando o funcionamento da função “toggle_notifications()”.
 - b. Identifique e destaque a(s) linha(s) no código responsável(is) por calcular o horário em que cada notificação será enviada.
 - c. Altere a(s) linha(s) identificadas no item 3b para que o método evite de usar um horário entre as 00:00 e 06:00h. Mantendo o número de notificações por dia solicitado e garantindo que cada notificação será dada em horários diferentes.

```
public void toggle_notifications() {
    int current_hour, notify_per_day;
    Date now = new Date();
    int notify_times[] = null;
    notify_per_day = radioButtonNotify.getindex();
    SimpleDateFormat dateFormatter = new SimpleDateFormat("h");
    current_hour = Integer.valueOf(dateFormatter.format(now).toString());
    if(not_per_day > 0){
        int div = 24/(1+notify_per_day);
        for (int i; i < notify_per_day; i++){
            current_hour += div;
            if (current_hour >= 24){
                current_hour -= 24;
            }
            notify_times[i] = current_hour;
        }
    }else{
        if (notify_times != null){
            Arrays.fill(notify_times,null);
        }
    }
    Register_cellphone_memory(notify_times);
}
```