Documentação do Trabalho 2 de Linguagem de Programação
Departamento de Ciência da Computação DCC – UFJF

Professor: Leonardo Vieira dos Santos Reis

Nome: Guilherme Fiorini Justen
Matrícula: 201965041AC

O fluxo principal da aplicação é feito em duas funções. A função main, e a função game.

Main

A função main é a função principal. Seu tipo é IO (). Ela é quem gera uma senha aleatória, e passa essa senha para a função game, onde o CriptoGame realmente vai acontecer. O usuário irá ficar na função game até vencer o CriptoGame, e depois voltará para a função main, onde será informado que venceu o jogo e verá quantas rodadas foram necessárias. Após, será perguntado pela função main se deseja jogar o jogo novamente ou não.

Game

A outra função do fluxo principal é a game. Nela é onde o CriptoGame está de fato. Ela recebe como parâmetro uma lista, que é a senha gerada aleatoriamente, e um contador, que inicialmente é igual a 1, e será incrementado a cada chamada da função, resultando no número de tentativas necessárias para vencer o jogo.

Cada vez que game é executada, uma tentativa é feita. Para isso, a função game deve então pegar o input do usuário, através da função get_input, e transformar este input em uma lista.

Depois, será calculado o número de acertos totais e parciais do usuário. Para calcular os acertos totais, a estratégia usada foi filtrar somente os elementos da senha e do input que são iguais posição a posição, e depois calcular o tamanho desta lista. Para filtrar foi usada a função filter, e para checar se um valor for igual a outro foi usada a função zipWith junto do operador (==).

Os acertos parciais foram calculados através da função get_partial_hits, explicada melhor na próxima seção. Ela retorna um inteiro, que é o número de elementos que não estão presentes na senha e no input do usuário. Este número quando subtraído de 4, que é o tamanho da senha e do input, representa a quantidade de elementos iguais

em ambas as listas. Ao final, ainda é necessário subtrair deste número a quantidade de acertos totais, para que de fato sejam contabilizados somente os acertos parciais.

Após o cálculo dos acertos parciais e totais, a função game vai informar ao usuário quantos acertos totais e parciais ele teve, e verificar se ele acertou a senha. Caso ele tenha a acertado, ele irá retornar para a função main com o número de tentativas necessárias para acertar, representado pela variável counter. Caso contrário, a função game será novamente chamada, passando a senha, inalterada, e a variável counter, incrementando-a em 1.

Para melhor legibilidade e redigibilidade do código, foram desenvolvidas funções auxiliares, descritas a seguir:

- get_input – Esta função é responsável por informar ao usuário que ele deve digitar uma senha de 4 dígitos por meio do caractere “?”, e por pegar o input do usuário. Ela também faz validação do input, caso o tamanho da lista formada pelo input não for 4, ou algum dos inteiros inseridos estiver fora do intervalo [1,6], o programa avisa ao usuário que ele deve inserir dados válidos, e disponibiliza o prompt novamente. Ela é feita com funções IO, e retorna uma lista de inteiros.

- get_partial_hits – Esta função auxiliar ajuda no cálculo do número de acertos parciais do usuário. Ela recebe duas listas, a senha e o input do usuário. A função tenta, para cada elemento do input, remover um elemento da senha, mas somente se ele estiver presente na senha. Assim, no final, somente ficarão na lista os elementos que não estão presentes no input, mas estão presentes na senha. Após a execução deste comando para cada elemento do input, ela retorna o tamanho da lista final com os elementos removidos.

- generate_password – Esta função é responsável por gerar uma senha aleatória, contendo 4 dígitos entre 1 a 6. Ela usa a função replicateM da biblioteca Control.Monad, e a função randomRIO da biblioteca System.Random. A replicateM replica um certo argumento x vezes dentro de uma lista. No meu caso, repliquei 4 vezes um número aleatório (gerado através do randomRIO). Com o uso do operador “$”, os 4 valores são colocados em uma lista.

- remove_elt – Esta função remove a primeira ocorrência de um elemento de uma lista. Ela recebe dois parâmetros, o valor do elemento a ser removido, e uma lista. Ela foi implementada usando casamento de padrão, guardas e recursão.

Interação com o usuário

Todo o código fonte está no arquivo index.hs, para executá-lo via terminal, basta estar na pasta do arquivo fonte e inserir o seguinte comando: “ghci index.hs”.

Após entrar no ambiente do ghci o usuário deve chamar a função main, digitando “main” no terminal. Depois disso, ele já será levado para o prompt onde deve tentar adivinhar a senha gerada pelo computador.