Algoritmo "Bubblesort"

Var

senha: vetor [1..20] de inteiro

* O vetor unidimensional é definido com 20 posições

elemento, i, fim, opcao, x, j, dado: inteiro

* Variáveis do tipo inteiro são declaradas

procedimento cadastro (nome: inteiro): inteiro

* Se inicia um procedimento

Var

Inicio

senha[elemento] := nome

* Se da o parametro

elemento <- elemento + 1

* a var elemento ganha +1 em seu valor

Fimprocedimento

* acaba o procedimento

funcao limpa():inteiro

* começa o procedimento (limpa)

var

resposta:inteiro

inicio

se i >= 1 entao

* se i for maior igual a 1 faça

resposta <- senha[i]

* resposta irá ganhar o valor de senha[i]

senha[i] <- 0

* senha[i] ganha o valor de 0

i <- i – 1

* subtrai 1 em i

elemento <- elemento – 1

* subtrai 1 em elemento

escreva ("Senha removida: ")

* recebe uma mensagem que sua senha foi removida

retorne resposta

* devolve a resposta ao seistema

Senao

escreval("Não existem senhas à serem removidas!")

* aparece na tela "Não existem senhas à serem removidas!"

fimse

fimfuncao

procedimento remove()

* cria o procedimento removeu

var

inicio

para i de 1 ate 20 faca

* repita i de 1 ate 20

senha[i] := 0

* senha[i] ganha o valor de 0

fimpara

i <- 0

* i recebe o valor de 0

elemento <- 1

* elemento tem seu valor definido como 1

fimprocedimento

procedimento listar()

* começa o procedimento listar

var

c:inteiro

* variável c é inteira

inicio

c <- i

* c ganha o valor de i

escreval("Lista de senhas cadastradas:")

* escreve na tela "Lista de senhas cadastradas:"

para i de 1 ate 20 faca

se senha[i] >= 1 entao

* se senha[i] for maior ou igual a 1 faça

escreval(senha[i])

* escreve na tela "Lista de senhas cadastradas:"

fimse

fimpara

i <- c

* i ganha o calor de c

fimprocedimento

funcao vazio():logico

* cria a função vazio com valor logico

v:logico

* cria a v do tipo logico

inicio

v <- falso

* v recebe o valor de falso

se i = 0 entao

v <- verdadeiro

* se i for = 0 então v será verdadeiro

senao

v <- falso

* v recebe o valor de falso

fimse

retorne v

* retorna v

fimfuncao

funcao ordenar():caracter

* cria a função ordenar do tipo caracter

var

inicio

elemento <- elemento - 1

* subtrai 1 no valor de elemento

para i de 1 ate elemento faca

* repete i de 1 até o valor de elemento

para j de i + 1 ate elemento faca

* repete j de 1 até o valor de elemento

se senha[i] > senha[j] entao

* se senha[i] for maior que senha[j] faça

x <- senha[i]

* x ganha o valor de senha[i]

senha[i] <- senha[j]

* senha[i] ganha o valor de senha[j]

senha[j] <- x

* senha[j] ganha o valor de x

fimse

fimpara

fimpara

escreval("Lista de senhas ordenadas:")

* escreve na tela "Lista de senhas ordenadas:"

para i de 1 ate elemento faca

* repetir i de 1 até o valor de elemento

escreval(senha[i])

* escreva na tela o valor de senha[i]

fimpara

elemento <- elemento + 1

* elemento ganha +1

fimfuncao

funcao maiormenor():inteiro

* cria função maiormenor do tipo inteiro

var

menor, maior:inteiro

* Declara variáveis do tipo inteiro

inicio

maior <- 0

* o valor de maior é definido como 0

menor <- 0

* o valor de menor é definido como 0

para i de 1 ate elemento faca

* repete i de 1 até o valor de elemento

se senha[i] > maior entao

* Se senha[i] for maior que maior faça

maior <- senha[i]

* Maior ganha o valor de senha[i]

menor <- senha[i]

* Menor ganha o valor de senha[i]

fimse

fimpara

para i de 1 ate elemento faca

* Repita i de 1 até o valor de elemento

se senha[i] >= 1 entao

* Se senha[i] for maior ou igual a 1 faça

se senha[i] < menor entao

* Se senha[i] menor que menor faça

menor <- senha[i]

* Menor ganha o valor de senha[i]

fimse

fimse

fimpara

escreval("Maior senha: ", maior)

* escreve na tela "Maior senha: ", e o valor de maior

escreval("Menor senha: ", menor)

* escreve na tela "Menor senha: ", e o valor de menor

escreval("")

* Não escreve nada na tela

Fimfuncao

Inicio

fim <- 1

* Fim ganha o valor 1

elemento <- 1

* Elemento ganha o valor de 1

enquanto fim <> 8 faca

* Enquanto fim for diferente de 8

escreval("Digite a opção desejada:")

escreval("")

escreval("1-Adicionar uma senha no cadastro")

escreval("2-Remover e retornar a senha do topo do cadastro")

escreval("3-Remover todas as senhas do cadastro")

escreval("4-Listar todas as senhas cadastradas")

escreval("5-Verificar se o cadastro está vazio")

escreval("6-Ordenar senhas em forma crescente")

escreval("7-Mostrar maior e menor senha cadastradas")

escreval("8-Sair do sistema")

escreval("")

* Menu de opções

leia (opcao)

* Le opção para fazer uma escolha

escolha opcao

* Realiza uma opção

caso 1

* Caso opção seja

escreval("Digite a senha à cadastrar:")

* Escreve na tela "Digite a senha à cadastrar:"

leia(dado)

* O usuário define o valor de dado

cadastro(dado)

* Chama o procedimento cadastro com o parâmetro dado

escreval("")

* Escreve na tela para pular uma linha

i <- i + 1

* Soma 1 a i

fim <- 1

* O valor de fim é definido como 1

caso 2

* Caso opção seja 2

escreval("")

* Escreve na tela para pular uma linha

escreval(limpa())

* Escreve na tela a mensagem removida

escreval("")

* Escreve na tela para pular uma linha

fim <- 2

* O valor de fim é definido como 2

caso 3

* Caso opção seja 3

se i >= 1 entao

* Se i for maior que 1 faça

escreval("")

escreval("Removendo todas as senhas da lista...")

escreval("")

escreval("Senhas removidas com sucesso!")

escreval("")

remove()

* Executa o procedimento de remover

senao

escreval("")

* Escreve na tela para pular uma linha

escreval ("Não existem senhas no cadastro à serem removidas!")

* Aparece na tela "Não existem senhas no cadastro à serem removidas!"

escreval("")

* Escreve na tela para pular uma linha

fimse

fim <- 3

* Fim tem seu valor definido como 3

caso 4

* Caso opção seja 4

escreval("")

* Escreve na tela para pular uma linha

se i >= 1 entao

* Se i for maior ou igual a 1 faça

listar()

* Executa o procedimento listar

senao

escreval("")

* Escreve na tela para pular uma linha

escreval ("Não existem senhas cadastradas para serem listardas!")

* Mostra na tela "Não existem senhas cadastradas para serem listardas!"

escreval("")

* Escreve na tela para pular uma linha

fimse

fim <- 4

* Fim tem seu valor definido como 4

caso 5

* Caso opção seja 5

escreval("")

* Escreve na tela para pular uma linha

escreval("Cadastro Vazio = ",vazio())

* Mostra na tela “cadastra vazio =” e o valor do procedimento vazio

escreval("")

* Escreve na tela para pular uma linha

fim <- 5

* fim tem seu valor definido como 5

caso 6

* Caso opção seja 6

fim <- 6

* Fim tem seu valor definido como 6

escreva(ordenar())

* Mostra na tela o valor da função ordenar

caso 7

* Caso opção seja 7

fim <- 7

* Fim tem seu valor definido como 7

escreva(maiormenor())

* Mostra na tela o valor da função maiormenor

caso 8

* Caso opção seja 8

fim <- 8

* Fim tem seu valor definido como 8

escreval("")

* Escreve na tela para pular uma linha

escreval ("Programa finalizado pelo usuário!")

* Mostra na tela “Programa finalizado pelo usuário!”

outrocaso

* Caso não seja escolhida nenhuma outra das opções está é escolhida

escreval("")

* Escreve na tela para pular uma linha

escreval("Opção inválida!")

* Aparece na tela “Opção inválida!”

escreval("Escolha uma opção válida abaixo para continuar!")

* Aparece na tela “Escolha uma opção válida abaixo para continuar!”

escreval("")

* Escreve na tela para pular uma linha

fimescolha

fimenquanto

Fimalgoritmo

* Todos os fim(algumacoisa) estão finalizando um comando no caso o fimalgoritmo está finalizando o código.