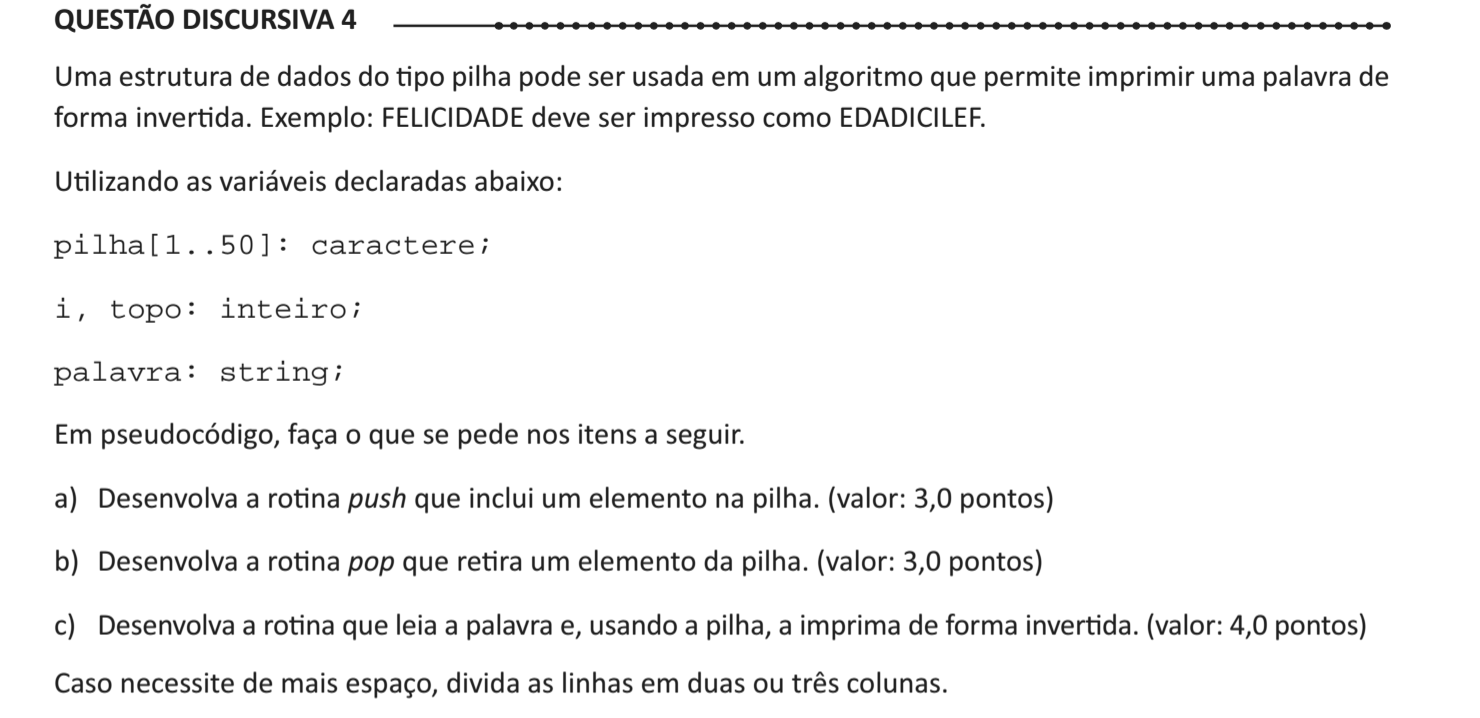
| L logo 25 anos.png  **Universidade Luterana do Brasil**  **ULBRA – Campus Torres**  **Pró-Reitoria de Graduação** | | Tipo de atividade:  Prova ( ) Trabalho ( ) ..... ( )  Avaliação: G1 ( ) G2 ( )  Substituição de Grau: G1 ( ) G2 ( ) | |
| --- | --- | --- | --- |
| Curso: | Disciplina: | | Data: |
| Turma: | Professor(a): | | Valor da Avaliação:  Nota: |
| Acadêmico(a): n°: | | |

Questão 01.



| push(char letra){ if(topo<50){ topo++;  pilha[topo]=letra; } } |
| --- |
|  |

| pop( ){  se (topo > 0){  topo-- }senao{  print("não é possivel"); } } |
| --- |

| reverter(String palavra){  for(int i=0;i<palavra.lenght;i++{  push(pop());}  }  char pop(){  if(topo>0){  topo--;  return getcharAt(topo+1); }  } |
| --- |
|  |

Questão 02.

Implemente um método para adicionar um elemento em uma pilha, considere que a pilha será implementada com um vetor de 10 elementos (variável global), e o topo da pilha é passado por parâmetro de referência para o procedimento.

| public static int vetor[10] = new int[10]; intt topo=-1; public static void push(int topo, int valor){  if(topo<10){ topo++; this.vetor[topo]=valor; }else{ sout( "Pilha cheia"); }     } |
| --- |

public class Pilha {

public static int[] *vetor* = new int[10];

public static int *topo* = -1;

public static void push(int valor) {

if (*topo* < 9) { // Verifica se a pilha não está cheia

*topo*++;

*vetor*[*topo*] = valor;

} else {

System.*out*.println("Pilha cheia");

}

}

public static void main(String[] args) {

// Exemplo de uso

*push*(5); // Adiciona o valor 5 à pilha

*push*(10); // Adiciona o valor 10 à pilha

// ...

}

}

Questão 03

Implemente um procedimento para remover um elemento de uma pilha, considere que a pilha será implementada com um vetor de 10 elementos, e o topo da pilha é passado por **parâmetro de valor** pro procedimento e **retornado** com um valor para o algoritmo

| public static int vetor[10] = new int[10]; int topo=-1; public static void pop (int topo){  if(topo>-1){ topo--; return topo }else{ sout( "Pilha vazia"); }  } |
| --- |
|  |
|  |

public class Pilha {

public static int[] *vetor* = new int[10];

public static int *topo* = -1;

public static int pop(int topo) {

if (topo > -1) { // Verifica se a pilha não está vazia

topo--;

return topo;

} else {

System.*out*.println("Pilha vazia");

return topo;

}

}

public static void main(String[] args) {

// Exemplo de uso

*topo* = *pop*(*topo*); // Remove um elemento da pilha

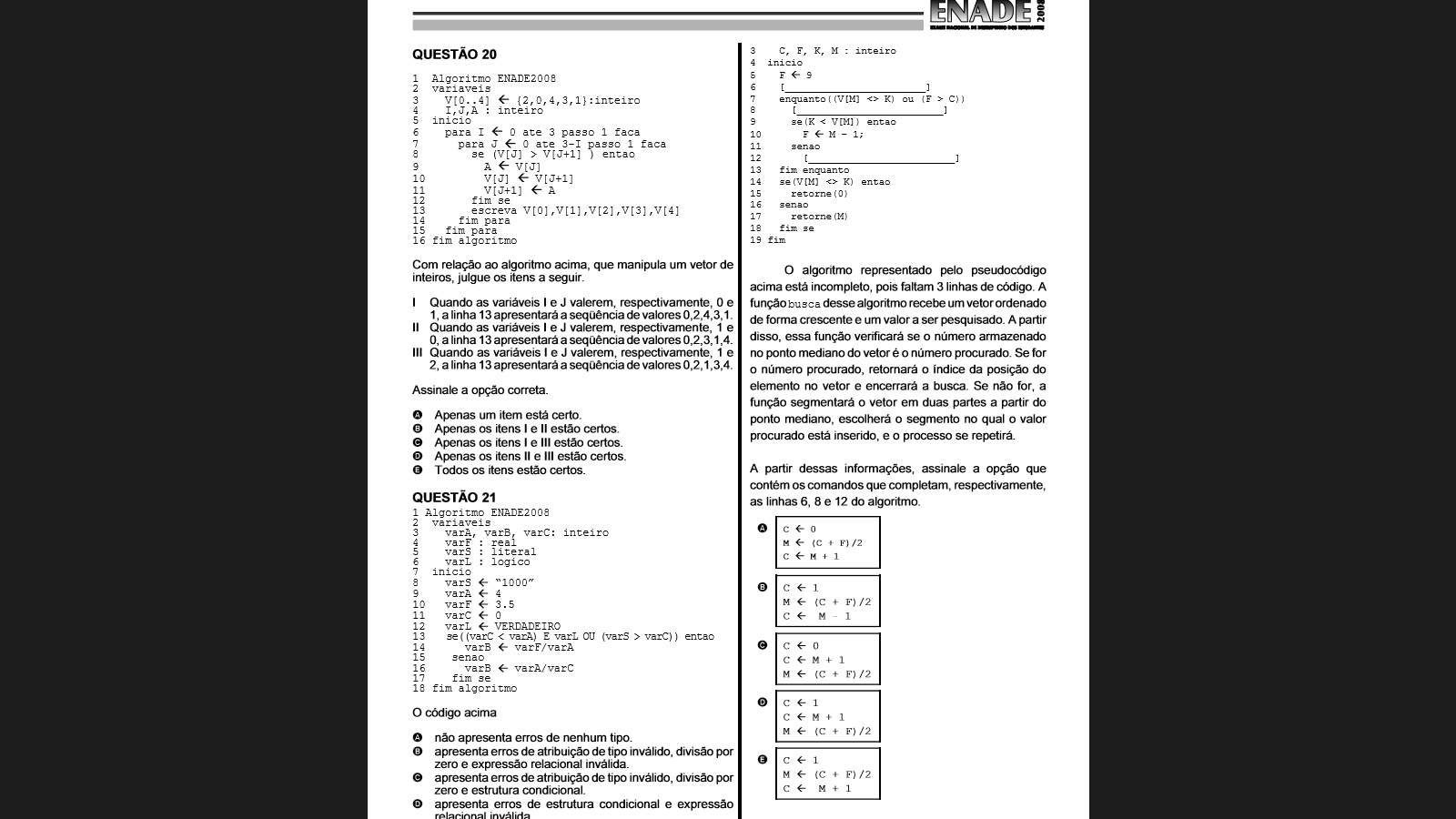
// ...

}

}

Questão 04.

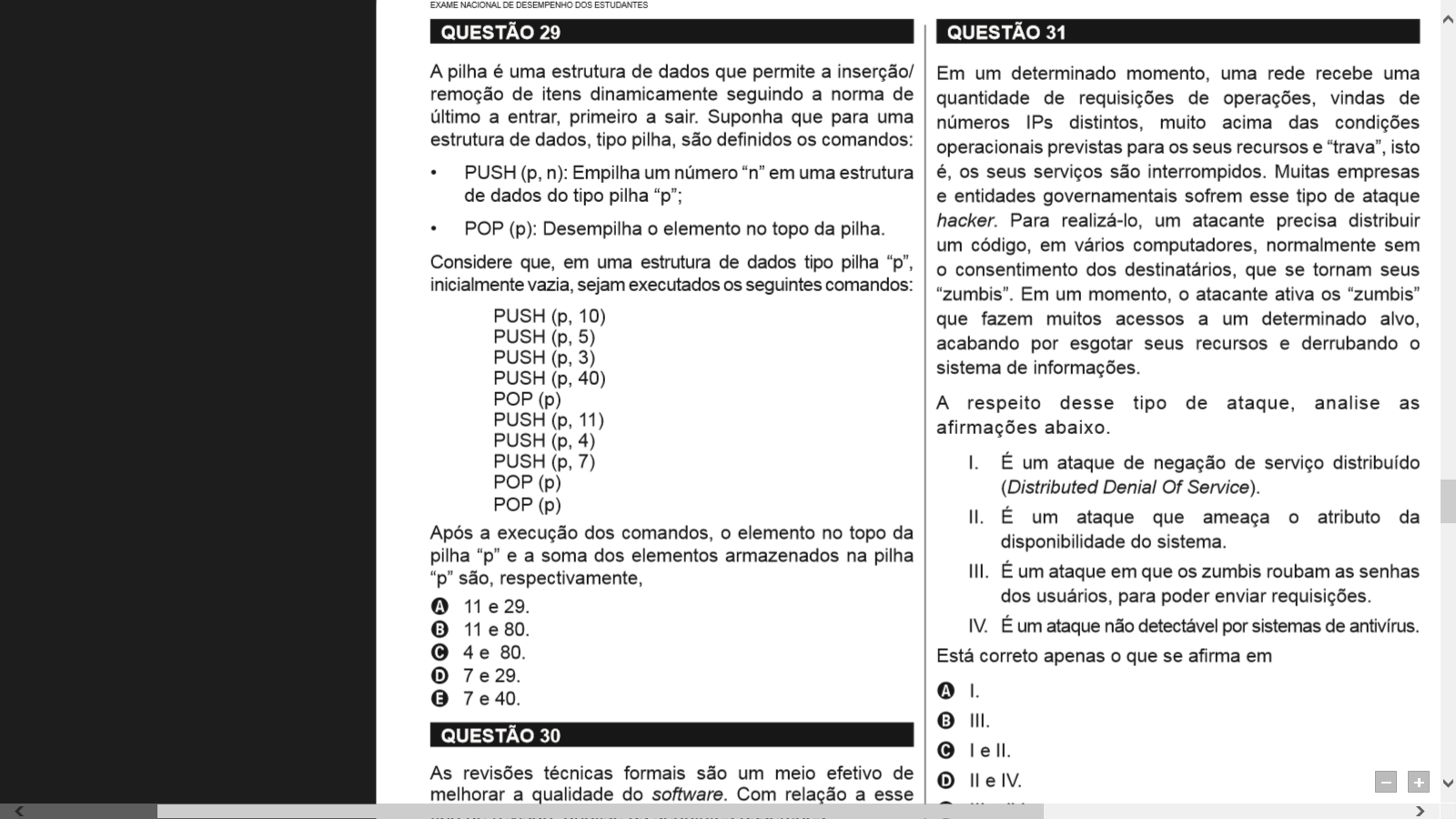
Essa questão foi retirada da prova do ENADE de 2008.



Letra E Todos os itens estão certos

Questão 05.

Essa questão foi retirada da prova do ENADE de 2011.



11 e 29

Questão 06.

Um entregador de pizzas deve entregar três pizzas em locais diferentes, o mesmo colocou as pizzas no baú conforme sua rota de entrega, a primeira pizza colocada no baú é a primeira pizza a ser entregue, ao chegar na primeira casa, o entregador se deu conta do seu erro. Qual foi o erro do entregador? Escreva sobre, abordando os conceitos de LIFO E FIFO.

O erro do entregador foi utilizar o conceito de LIFO (Last In, First Out), em vez de FIFO (First In, First Out), ao organizar as pizzas no baú de entrega.

Questão 07.

Conforme o trecho de código abaixo, qual possui melhor desempenho em relação ao número de comparações? E qual realizara o menor número de trocas? Justificar por texto ou por testes de mesa.

