UNIARA Centro Universitário de Araraquara

Curso: Engenharia de Computação e Sistemas de Informação

Disciplina: Estruturas de Dados

Avaliação 2º B

Nome: _____ Matrícula: _____

QUESTÕES

1) (3,0) Uma pilha é uma estrutura de dados que armazena uma coleção de itens de dados relacionados e que garante o seguinte funcionamento: o último elemento a ser inserido é o primeiro a ser removido. É comum na literatura utilizar os nomes *push* e *pop* para as operações de inserção e remoção de um elemento em uma pilha, respectivamente. O seguinte trecho de código em linguagem C define uma estrutura de dados pilha utilizando um vetor de inteiros, bem como algumas funções para sua manipulação.

```
#include <stdlib.h>
#include <stdio.h>
typedef struct {
        int elementos[100];
        int topo;
}pilha;
pilha * cria_pilha() {
        pilha * p =malloc(sizeof(pilha));
        p->topo = -1;
        return pilha;
}
void push(pilha *p, int elemento) {
        if (p->topo >= 99)
                return;
        p->elementos[++p->topo] = elemento;
}
int pop(pilha *p) {
        int a = p->elementos[p->topo];
        p->topo--;
        return a;
}
//O programa a seguir utiliza uma pilha
void main() {
        pilha * p = cria_pilha();
        push(p, 2);
        push(p, 3);
        push(p, 4);
        pop(p);
        push(p, 2);
        int a = pop(p) + pop(p);
        push(p, a)
        a += pop(p);
        printf("%d", a);
}
```

A esse respeito, avalie as afirmações a seguir:

I. *O valor exibido pelo programa seria o mesmo caso a instrução a += pop(p); fosse trocada por a += a;*

II. *Em relação ao vazamento de memória (memory leak), é opcional chamar a função free(p), pois o vetor usado pela pilha é alocado estaticamente.*

É correto o que se afirma em:

a) I, apenas
b) II, apenas
c) I e II
d) Nenhuma das anteriores.

UNIARA Centro Universitário de Araraquara

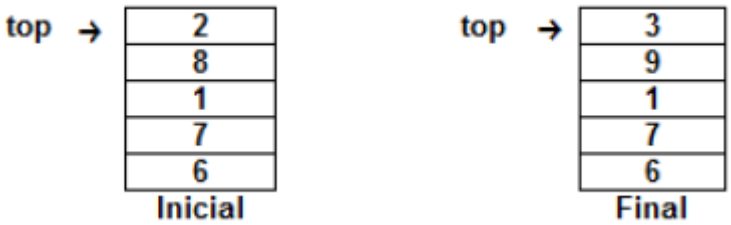
Curso: Engenharia de Computação e Sistemas de Informação

Disciplina: Estruturas de Dados

Avaliação 2º B

Nome: _____ Matrícula: _____

2) (2,0) Considere os estados (inicial e final) da pilha a seguir, na qual top corresponde ao topo da pilha.



Para atingir o estado final dessa pilha, deve-se usar a seguinte sequência de operações básicas:

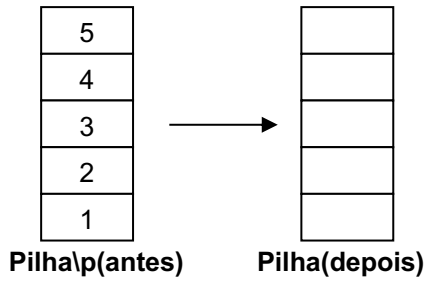
a) pop(), pop(), push(9), push(3).
b) push(2), push(8), pop(), pop().
c) push(), push(), pop(8), pop(2).
d) pop(3), pop(9), push(), push().
e) Nenhuma das anteriores.

3) (3,0) Considerando o programa em anexo, operações sobre Pilhas, preencha como a pilha ficará após o final da função interrogação.

```
void interrogacao(tpilha *pp){
    tpilha aux;
    criar(&aux);
    telemento elemento_x, elemento_y, valor;

    if(pop(pp, &elemento_x)){
        printf("O que aconteceu? \n");
    }
    while((*pp)->prox != NULL){
        if(pop(pp, &valor)){
                printf("O que aconteceu? \n");
         }
         if(push(&aux, valor)){
                printf("O que aconteceu? \n");
         }
    }
    if(pop(pp, &elemento_y)){
        printf("O que aconteceu? \n");
    }

    if(push(pp, elemento_x)){
        printf("O que aconteceu? \n");
    }
    while(aux != NULL){
        if(pop(&aux, &valor)){
          printf("O que aconteceu? \n");
         }
        if(push(pp, valor)){
          printf("O que aconteceu? \n");
        }
    }
     if(push(pp, elemento_y)){
        printf("O que aconteceu? \n");
     }
}
```



UNIARA Centro Universitário de Araraquara

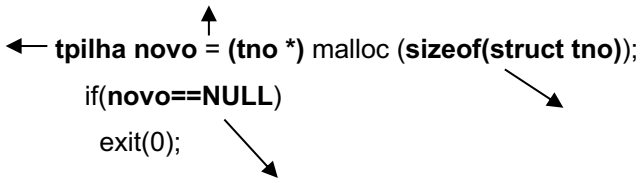
Curso: Engenharia de Computação e Sistemas de Informação

Disciplina: Estruturas de Dados

Avaliação 2º B

Nome: _____ Matrícula: _____

4) (2,0) Explique o que é a função malloc e qual é a biblioteca que a fornece em C. Seguindo as setas, explique o que cada "comando" ou "instrução" faz no trecho do programa abaixo. Considere "novo" como uma variável ponteiro para estrutura de pilha.



Anexo:

```
#include<stdio.h>
#include<stdbool.h>
#include<stdlib.h>

typedef int telemento;

typedef struct no{
       int dado;
       struct no * prox;
}tno;

typedef tno * tpilha;

void criar(tpilha * pp){
    *pp = NULL;
}

bool push(tpilha *pp, telemento valor){
    tpilha novo;
    novo = (tno*) malloc(sizeof(tno));
    if(novo == NULL)
      return false;
    else{
        novo->dado = valor;
        novo->prox = *pp;
        *pp = novo;
        return true;
    }
}
```

```
bool pop(tpilha *pp, telemento * valor){
    tpilha aux;
    if(*pp == NULL)
      return false;
    else{
        aux = *pp;
        *valor = (*pp)->dado;
        *pp = aux->prox; //*pp = *pp->prox;
        free(aux);
        return true;
    }
}

void main(){
    tpilha p;
    int op, valor = 0, pegaValor = 0;
    criar(&p);

    do{
      printf("Entre com  4- Interrogacao ou (-1 Sair): ");
      scanf("%d", &op);
      switch(op){
              …
              case 1: interrogacao(&p);
                    break;
      }
    }while(op!=-1);
    system("PAUSE");
}
```