Exercício Cap1

Alunas: Gabriela Duarte Albornoz e Gilderlane de Alencar Sousa.

1.1 Analise a estrutura de dados esquematizada a seguir:



Explique por que essa estrutura:

( a ) Não é um conjunto.

Resposta: Pois há ordem e repetição.

( b ) Não é uma lista.

Resposta: Não uma lista, pois não é uma coleção de itens organizados linearmente

(c ) Não é uma árvore.

Resposta: Pois não é organizada hierarquicamente.

1.2 Indique a estrutura de dados mais adequada para ser usada num programa que precisa representar:

( a ) Estradas entre cidades de um mapa.

Resposta: Grafo, pois as cidades se relacionam através das estradas de forma parecida a um grafo.

( b ) A estrutura organizacional de uma empresa.

Resposta: Árvore, pois esta organizado de forma hierárquica

( c ) A fila de impressão em um sistema operacional.

Resposta: Lista, pois é sequencial.

( d ) A estrutura de diretórios em um sistema operacional.

Resposta: Árvore.

1.3 Ao ser executado, o programa a seguir exibe a palavra diferentes. Explique por que isso acontece.

#include <stdio.h>

int main(void) {

char s[3] = "um";

char t[3] = "um"*;*

if*(*s == t ) puts ("iguais")*;*

else puts ("diferentes")*;*

return 0;

}

Resposta: As variáveis são ponteiros que estão apontando para o mesmo local de memória.

1.4 Explique o que acontece quando o programa a seguir é executado.

#include <stdio.h>

typedef struct { char valor[10]; } Str;

int main(void) {

Str x = { "um" };

Str y = { "dois" );

puts (x.valor)*;*

*x = y;*

puts (x.valor)*;*

return 0*;*

*}*

Resposta: O programa irá exibir as palavras “um” e “dois”.

1.5 Explique o que acontece quando o programa a seguir é executado.

#include <stdio.h>

int main(void) {

int a = 3, b = 5*;*

int \*p = &a, \*q = &b;

\*P = \*p + \*q;

\*q = \*p - \*q*;*

\*p = \*p - \*q*;*

printf("%d, \*d\n", a, b)*;*

return 0;

}

Resposta: O resultado que será exibido é 5 e 3.

1.6 Indique a forma de alocação de memória (estática sequencial, estática encadeada, dinâmica sequencial ou dinâmica encadeada) usada para armazenar os itens 1, 2, 3 e 4, em cada um dos programas a seguir.

(a) #include <stdio.h>

#include <stdlib.h>

int main(void) {

int i, \*v = malloc (4\*sizeof(int)*)*;

for*(*i=0*;* i<4; i++) v[i] = i+1;

for(i=0; i<4; i++)

printf("%d\n", v[i]);

return 0*;*

*}*

Resposta: Dinâmica sequencial.

(b) #include <stdio.h>

typedef struct no {

int item;

struct no \*prox*;*

} No;

int main(void) {

No v[4] = {{3,v+2}, {1, v+3}, {4, NULL}, {2, v+0});

for (No \*p = v+1; p != NULL; p = p->prox)

printf("%d\n", p->item);

return 0*;*

*}*

Resposta: Estática encadeada.

(C) #include <stdio.h>

#include <stdlib.h>

typedef struct no {

int item;

struct no \*prox;

} \*Lst;

Ist no(int x, Lst p) {

Lst n = malloc(sizeof(struct no)*);*

n->item = x*;*

n->prox = p*;*

return n*;*

int main(void) {

Lst p = no(1, no (2, no (3, no (4, NULL)))*);*

while( p ) {

printf("ed\n", p->item);

p = p->prox*;*

return 0;

}

}

Resposta: Dinâmica encadeada.