Nome:  
  
**Revisão de Estrutura de Dados.**

1) Qual a ideia do uso de Struct e onde podemos utilizar esta estrutura de dados?

Uma struct é uma maneira de juntar várias variáveis em um só local, podemos usar para organizar formulários, fichas ou até mesmo outras estruturas de dados

2) Como definimos uma Struct?

typedef struct{

int exemplo1;

char exemplo[2];

float exemplo3;

}nomedastruct;

3) Qual a definição da Struct para criação das estruturas abaixo:

a) Pilha

#define TAM 10

typedef struct{

int vet[TAM];

int topo

}pilhavet;

b) Fila

#define TAM 10

typedef struct{

int R,F;

int vet[TAM];

}filavet;

#define TAM 10

c) Fila Circular

typedef struct{

int R,F,Q;

int vet[TAM)];

}filacircular;

4) Observe a de definição de código a seguir: (1,0 ponto)

tipo tipest{

int Código;

char Descricao;

float Preco;

};

De acordo com a definição, tipest é do tipo:

Resposta:

A ( ) – Variável composta homogênea.

B ( ) – Virtual.

C ( ) – Struct.

D ( ) – Vetor.

E ( ) – Matriz.

5) Em relação aos conceitos e operadores usados em alocação estática e dinâmica, assinale a afirmação correta:

Resposta:

A ( ) - Alocação dinâmica permite aumentar a quantidade de memória alocada, mas não permite a redução.

B ( ) – Para alocar memória dinamicamente, é necessário antes, separar o espaço de memória que será disponibilizado para o recurso.

C ( ) – Alocação estática permite apenas o uso de arrays para interagir com a memória.

D (X) – A alocação dinâmica poupa memória embora gere maior gasto de processamento.

E ( ) – Função sizeof determina o número de bits para um determinado tipo de dados.

6) A alocação de memória é dividida em partes. Qual decisão da alocação é tomada quando o programa é compilado?

Resposta:

A ( ) - Alocação Dinâmica;

B (X) - Alocação Estática;

C ( ) - Alocação Híbrida;

D ( ) - Alocação Local;

E ( ) - Em todas as partes.

7) São operações válidas ao se trabalhar com structs:

1. Atribuição de variáveis da estrutura a variáveis da estrutura de mesmo tipo.
2. Comparar estruturas usando == e !=
3. Coleta de endereço de uma variável de estrutura (operador &)
4. Acesso aos membros de uma variável de estrutura

Estão corretos os itens:

Resposta:

A ( ) - I e II.

B ( ) - I e III

C ( ) - II, III e IV

D ( ) - I, III e IV

E ( ) - Todas os itens estão corretos.

8) Implemente um programa em C que armazene informações de um conjunto fixo de n (passando pelo usuário) funcionários de uma empresa. O programa deverá solicitar ao usuário o nome, idade e salário de cada funcionário. Ao final, o programa deve exibir as informações de todos os funcionários e calcular a média salarial da equipe.

9) Qual a diferença em uma pilha e uma fila?

Pilha usa FILO e a fila FIFO

10) Qual a função da variável topo na pilha?

Pegar o dado que está no topo

11) Qual a função das variáveis R e F em uma fila?

A R verifica se está Cheia e a F verifica se está vazia

12) Qual o trecho do código que temos o empilhamento de um valor?

void empilha(int valor, pilhavet \*p){

if(p -> topo == TAM-1){

printf(“\nPilha Cheia”);

exit(0);

}

p -> topo++;

p -> vet[p -> topo]=valor;

}

13) Qual o trecho do código que temos o desempilhamento de um valor?

int desempilha(pilhavet \*p){

int aux;

if(p -> topo <0){

printf(“\npilha vazia”);

exit(1);

}

aux = p ->vet [p -> topo];

p -> topo--;

return aux;

}

14) Qual o trecho do código que temos o enfileiramento de um valor?

void enfileirar(int valor, filavet \*p){

if (p -> R == TAM-1){

printf(“\n Fila Cheia!”);

exit(0);

}

p -> R++;

p -> vet[p -> R] = valor;

}

15) Qual o trecho do código que temos o desenfileiramento de um valor?

int desenfileirar (filavet \*p){

int aux;

if (p -> F == TAM){

printf(“\nFila Vazia!”);

exit(1);

}

aux = p -> vet[p -> F];

p -> F++;

return aux;

}

16) Qual as características que diferem uma fila de uma fila circular?

A fila circular permite a realocação de memória, evitando desperdício

17) Qual a saída do código abaixo:

  int a, b;

    int \*ponteiro = &a;

    a = 40;

    b = 60 + 20;

    \*ponteiro = b\*3;

    printf("%d", a);

40

18) O que Listas Encadeada? Como é feito a alocação neste estrutura. (Pesquise!)

19) Explique o que é a função malloc em C, descrevendo seu funcionamento, em quais situações ela deve ser utilizada e o que ocorre quando a alocação de memória falha

20) Considere que você deseja criar um programa em C para armazenar uma lista de números inteiros. No entanto, a quantidade de números a ser armazenada só será conhecida durante a execução do programa, quando o usuário informar a quantidade desejada. Para isso, utilize a função malloc para alocar dinamicamente um array que armazene os números inteiros.

O programa deve:

1. Solicitar ao usuário a quantidade de números que ele deseja armazenar.
2. Alocar memória dinamicamente utilizando malloc para o array de inteiros.
3. Solicitar ao usuário para preencher cada um dos números.
4. Exibir os números digitados.
5. Liberar a memória alocada antes de encerrar o programa.