Nome: Caio Pita Gama \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Revisão de Estrutura de Dados.**

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

A ideia da struct é um único dado com várias variáveis que podemos utilizar em fila, pilha, fila circular, ect.

2). Como definimos uma Struct?

Definimos uma struct com o typedef Struct, abrimos {} e dentro deles, colocamos as variáveis que farão parte dessa struct.

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

a) Pilha

É uma estrutura de dados que inicialmente foi feita de forma sequencial (vetor). Hoje temos pilha dinâmica (usando encadeamento).Sege o critério LIFO( Last in First out) (Último a entrar Primeiro a sair)

b) Fila

Fila é uma estrutura de dados que inicialmente foi feita de forma sequencial (vetor), hoje temos lista encadeada. Segue o critério FIFO( First in First out) ( Primeiro a entrar é o primeiro a sair).

c) Fila Circular

Bem parecido com a fila, mas com a diferença de algumas melhorias. Quando a fila ficava cheia e vazia, as variáveis R e F ficavam no topo da fila e não conseguimos alocar novos conteúdos, desperdiçando memória, então na fila circular temos uma terceira variável, que resolve esse problema, chamada de variável auxiliar, que ajuda a conseguimos alocar novos elementos na lista.

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 ( x ) – 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 ( ) – 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 ( x ) - 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?

Na pilha o último a entrar é o primeiro a sair e na fila o primeiro a entrar é o primeiro a sair.

10). Qual a função da variável topo na pilha?

A função da variável topo na pilha é para saber qual é o elemento que está no topo da pilha

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

R – Coloca as informações na Fila

F – Tira as informações da Fila

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

for(int i=0;i<TAM;i++){

printf("\nDigite o valor a ser empilhado:\n");

scanf("%d",&valor);

empilha(valor,&pilha);

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

for(int i=0;i<TAM;i++){

printf("\n%d",desempilha(&pilha));

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

for(int i=0;i<TAM;i++){

printf("\nDigite o valor a ser enfileirado: \n");

scanf("%d",&valor);

enfileirar(valor,&fila);

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

for(int i=0;i<TAM;i++){

printf("\n%d",desenfileirar(&fila));

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

Na fila circular temos uma variável auxiliar que ajuda para conseguirmos inserir novos valores após a fila ser esvaziada.

17). Qual a saída do código abaixo:

  int a, b;

    int \*ponteiro = &a;

    a = 40;

    b = 60 + 20;

    \*ponteiro = b\*3;

    printf("%d", a);

Saida: 40

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

Uma lista encadeada é uma representação de uma [sequência](https://www.ime.usp.br/~pf/algoritmos/aulas/footnotes/math.html#sequence) de objetos, todos do mesmo tipo, na memória RAM (= *random access memory*) do computador. Cada elemento da sequência é armazenado em uma célula da lista: o primeiro elemento na primeira célula, o segundo na segunda, e assim por diante. Para estes casos pode-se alocar a memória necessária via alocação dinâmica de memória.

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.

A função malloc é usada para alocar memória dinamicamente durante a execução do programa. Ela permite que você aloque uma quantidade específica de memória conforme necessário, em oposição a alocação estática de memória, que é feita em tempo de compilação. Malloc é útil quando não se sabe quantas variáveis, objetos e espaço de memória será necessário. Aplicativos que alocam memória com frequência podem sofrer erros aleatórios de "fora da memória". Esses erros podem resultar em outros erros ou comportamento inesperado em aplicativos afetados.

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.

