

Dredd - Juiz Online

Principal

Perfil

Minhas Provas

Sair

Minutos
Restantes:
606

Usuário:
Gabriel Nathan
Almeida Silva

Notas:
Q1: ?
Q2: ?
Q3: 100
Total: 33

Quarta Lista Avaliativa

Prova Aberta Até: 11/12/2019 23:59:59**Número Máximo de Tentativas:** 3**Atenuação da Nota por Tentativa:** 0%

Instruções para a prova: A prova é individual. Desligue seu celular. Não converse com os colegas. Não fique olhando para a tela dos colegas.

Questão 1: Ponteiro - Correção de Código (2)

Faça uma análise e correção do código a seguir. O programa contém erros propositais, cabendo a você corrigi-los, conforme orientação no código. O código será corrigido manualmente pelos professores.

```
#include
using namespace std;
// CORRIJA -> registro composto de um campo inteiro e um ponteiro para um outro registro do mesmo tipo
struct noh{
    int dado;
    noh* &prox;
};
int main() {
    int i, total, soma=0;
    int ptrV[5] = {5,4,3,2,1};

    // CORRIJA -> aloca um vetor de registro dinamicamente
    int* vetor = new noh[5];
    i=0;
    while(i < 5) {
        vetor[i].dado = ptrV[i];
        soma+=vetor[i].dado;
        if (i != 4){
            vetor[i].prox = ptrV[i+1];
        } else{
            vetor[i].prox = NULL;
        }
        i++;
    }

    int *ptrValor = &total;

    // CORRIJA -> atribui o dobro de soma ao valor apontado por ptrValor
    // (não é permitido usar soma diretamente)
    ptrValor = soma*2;

    // CORRIJA -> imprime o valor apontado por prox em cada posição e o valor de total (indiretamente)
    // (note: imprime valor apontado, não o endereço).
    // Se não for possível imprimir o valor apontado por prox, imprima 0
    cout << vetor[i].prox << " " << *ptrValor<<"endl;

    // CORRIJA -> desaloca objetos alocados dinamicamente
    // (apenas objetos alocados dinamicamente devem ser desalocados)
    delete ptrValor;
    delete vetor[i].prox;
    delete ptrV;
}
```

Exemplo de saída:

```
4 30
3 30
2 30
1 30
0 30
```

Peso: 1

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

Minutos
Restantes:
606

Usuário:
Gabriel Nathan
Almeida Silva

Notas:
Q1: ?
Q2: ?
Q3: 100
Total: 33

Questão 2: Ponteiro e Arquivo Binário - Vetor dinâmico salvo em arquivo

Faça um programa que guarda números num vetor dinâmico. Não se sabe quantos números deverão ser armazenados até que se saiba quais são os números.

A estratégia de alocação de memória deve ser a seguinte:

- o vetor é criado inicialmente com capacidade 5,
- conforme é feito o armazenamento de números no vetor, novos espaços de memória vão sendo alocados, sempre com 5 elementos a mais do que antes.

O programa deverá ler números inteiros positivos de um arquivo binário chamado dados.bin e depois da entrada padrão. O último número em cada entrada de dados será um número não positivo, indicando o fim dos números. Considera-se que o zero não é positivo, conforme pode ser visto no exemplo de entrada.

Ao final da entrada de dados, o programa deverá:

1. Escrever todos os números (não negativos) armazenados.
2. Escrever qual a capacidade atual do vetor.
3. Escrever o número de transformações (realocações na memória) necessárias.
4. Atualizar o arquivo, gravando as alterações (**os valores devem ser escritos em formato binário.**).

Exemplo de Entrada:

(dados.bin):

10 18 15 71 42 41 0

(entrada padrão):

20 18 35 14 13 0

Exemplo de Saída:

10 18 15 71 42 41 20 18 35 14 13
15
2

Peso: 1

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

Questão 3: Ponteiros - Concatenação de dois vetores

Escreva um programa que leia duas cadeias de caracteres e concatene a segunda cadeia ao final da primeira. As duas cadeias de entrada devem ser declaradas como vetores de char com tamanhos máximos fixos, limitadas a 100 caracteres cada. A cadeia resultante deve ser declarada como um ponteiro para o tipo de dado char que aponta para uma variável dinâmica. Soluções usando a classe `string` serão desconsideradas.

Calcule a quantidade de memória necessária para armazenar o resultado da concatenação antes de alocar memória para o resultado. Obs: Em C++, a função `strlen` da biblioteca `cstring` pode ser usada para calcular o número de caracteres úteis de um vetor de char. Você pode utilizar as funções `strcpy`, `memcpy` ou similares para resolver o problema, caso acho interessante.

Entradas:

1. Elementos do primeiro vetor
2. Elementos do segundo vetor

Saídas:

- Vetor resultante da concatenação do primeiro vetor com o segundo

Exemplo de Entrada:

q w e r t y u i o p
m n b v c x z

Exemplo de Saída:

q w e r t y u i o p m n b v c x z

Minutos
Restantes:
606

Usuário:
Gabriel Nathan
Almeida Silva

Notas:
Q1: ?
Q2: ?
Q3: 100
Total: 33

Peso: 1

Última tentativa realizada em: 11/12/2019 13:53:51

Tentativas: 1 de 3

Nota (0 a 100): 100

Status ou Justificativa de Nota: Nenhum erro encontrado.

[Ver Código da Última Tentativa](#)

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

[Escolher arquivo](#)

Nenhum arquivo selecionado

[Enviar Resposta](#)



Desenvolvido por Bruno Schneider a partir do
programa original (Algod) de Renato R. R. de
Oliveira.

