ACADEMIC 1030

SUBMISSÕES | DESCRIÇÃO

ESTRUTURAS E BIBLIOTECAS | NÍVEL 5 | BASE TIME LIMIT: 1 SECOND | MEMORY LIMIT: 200 MB | THIS PROBLEM WAS CREATED BY ANDRÉ REIS DE BRITO

URI Online Judge | 1030

Lista encadeada

Por André Brito, Brazil

Timelimit: 1

Listas encadeadas são muito úteis, pois organizam de forma dinâmica um conjunto de informações. Através de listas encadeadas podemos fazer algumas operações que são mais complexas com vetores, como a remoção de elementos, ordenação e inserção de múltiplos elementos na lista.

Sua tarefa é implementar um programa que trabalhe com uma lista encadeada e receba um conjunto de números positivos de tamanho N qualquer. A leitura destes elementos é encerrada quando você recebe um número negativo. Em seguida, serão fornecidos alguns comandos e seus respectivos parâmetros. São eles

R - Remover o primeiro elemento x da lista. Se o elemento x não existir na lista, não faço nada. Recebe apenas um parâmetro: o valor de x. Ex: R x

I - Inserir um valor x na lista imediatamente antes do primeiro item y. Se o valor y não existir na lista, o valor x é inserido no final da lista. Recebe dois parâmetros: o valor de x e de y. Ex: I y x

X - Informa o fim do programa. Não recebe parâmetros.

Quando finalizar a execução, o programa deverá exibir a sequência de números respeitando a ordenação da lista após as inserções e remoções. Os números devem ser separados por um espaço em branco, e não se esqueça de quebrar a linha ao terminar de imprimir todos os números, caso contrário você pode receber um "presentation error".

OBS: Respostas que apresentem um resultado correto sem implementar listas encadeadas serão rejeitadas.

Entrada

Um conjunto de números inteiros de tamanho N desconhecido. Em seguida, serão enviados os comando R, I ou X, seguidos de seus respectivos parâmetros.

Saída

Os elementos da lista na ordem que estão salvos na lista, separados por espaços em branco.

OBS: Cada saída é encerrada sempre com um espaço em branco. Então se o resultado for 4 3 2 1 , teremos 4[espaço]3[espaço]2[espaço]1[espaço].

Samples Input	Samples Output
1 2 3 4 5 -1 R 4 I 5 6 R 5 I 5 9 X	6 3 2 1 9
1 2 3 4 -1 I 1 5 X	4 3 2 5 1
1 -1 I 1 2 I 2 3 I 3 4 I 4 5 I 5 6 R 1 X	6 5 4 3 2
1 2 3 4 5 6 -1 I 9 7 R 8 X	6 5 4 3 2 1 7

CUSTOM PROBLEM: 1030

LINGUAGEM: C++17

SOURCE CODE

```c
#include <stdio.h>

int main() {

    /**
     * Escreva a sua solução aqui
     * Code your solution here
     * Escriba su solución aquí
     */

    return 0;
}
```

CONSTRUA A SUA SOLUÇÃO E ENVIE!

ENVIAR

ACADEMIC 1031

SUBMISSÕES | DESCRIÇÃO

ESTRUTURAS E BIBLIOTECAS | NÍVEL 3 | BASE TIME LIMIT: 1 SECOND | MEMORY LIMIT: 200 MB | THIS PROBLEM WAS CREATED BY ANDRÉ REIS DE BRITO

URI Online Judge | 1031

Carga de um avião

Por André Brito, Brazil

Timelimit: 1

A CharlieAir é a maior empresa de transporte aéreo do país e seus aviões transportam importantes cargas de norte a sul. Entretanto, calcular a quantidade máxima de carga que seus aviões podem transportar nunca é uma tarefa fácil. São cálculos e mais cálculos para saber se uma simples caixa de 10kg vai poder embarcar ou não. Se o avião ultrapassar a quantidade máxima de peso, ele pode nem conseguir decolar, o que resultaria numa catástrofe!

Diante deste problema, um funcionário percebeu que a carga em gramas que o avião poderia carregar era equivalente a quantidade de memória livre em bytes disponível no computador de bordo da aeronave. Essa incrível coincidência fez com que ele pedisse para que a equipe de TI, da qual você faz parte, criasse um programa para ser executado no próprio avião.

Ciente de que o avião cargueiro mais antigo da frota é um avião da década de 90, sua equipe decidiu criar um programa em C que recebe um conjunto de números positivos que representam o peso de cada item que será transportado no avião.

Se o programa receber o valor 0 (zero) e ainda houver memória livre no computador de bordo, você deve exibir a mensagem "Pronto para decolar!". Caso a memória tenha acabado, exiba a mensagem "Sobrepeso!". A mensagem deve ser exibida no primeiro item que não pôde ser carregado no avião, encerrando o programa.

OBS: Soluções sem a implementação de ponteiros não serão consideradas!

Entrada

Sequência de números positivos N, 1 < N < 2.147.483.648, indicando o peso da carga em gramas e 0 quando terminar de inserir objetos no avião cargueiro.

Saída

"Pronto para decolar!", caso o avião esteja abastecido abaixo do peso máximo de decolagem e "Sobrepeso!", caso contrário.

Teste de sobrepeso

Cada computador terá uma saída diferente para um sobrepeso. Para testar, recomendo pegar alguns números aleatórios no link abaixo até que caia na situação de sobrepeso.

https://www.random.org/integers/?num=200&min=2500000&max=100000000&col=1&base=10&format=html&rnd=new

Samples Input	Samples Output
10101382 11136090 8460809 90142403 86083973 0	Pronto para decolar!
19464648 10919966 81153503 71754300 10101382 0	Pronto para decolar!

CUSTOM PROBLEM: 1031

LINGUAGEM: C++17

SOURCE CODE

```
#include <iostream>

using namespace std;

int main() {

    /**
     * Escreva a sua solução aqui
     * Code your solution here
     * Escriba su solución aquí
     */

    return 0;
}
```

CONSTRUA A SUA SOLUÇÃO E ENVIE!

ENVIAR

ACADEMIC 1032

SUBMISSÕES | DESCRIÇÃO

ESTRUTURAS E BIBLIOTECAS | NÍVEL 3 | BASE TIME LIMIT: 1 SECOND | MEMORY LIMIT: 200 MB | THIS PROBLEM WAS CREATED BY ANDRÉ REIS DE BRITO

URI Online Judge | 1032

Lista de pessoas

Por André Brito, Brazil

Timelimit: 1

Complete o programa a seguir, escrevendo a função "insere_pessoa" para incluir um novo conjunto de informações, ao final da lista. Observe que alguns campos ou trechos de código foram omitidos.

Considere que:

- a lista já contém alguns dados inseridos;

- a função recebe os dados novos através de um ponteiro para os dados de uma pessoa;

- a função recebe como parâmetro o ponteiro para o início da lista;

- utilize alocação dinâmica de memória.

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int insere_pessoa(PESSOA *p_head, struct_PESSOA *p_novo) {

   /* Há um trecho de código aqui */
  return0;

}

int main(void) {

  PESSOA *p_head, *p, pessoa1;

  p_head = NULL;

  if ((p = (PESSOA *)malloc(sizeof(PESSOA))) == NULL) {

    printf("Falta Memoria \n");

    return 1;

}

  /* insere os 1os dados na lista */

  strcpy (p -> nome, "Pessoa1");

  p -> sexo = 'F';

  p -> idade = 42;

  p -> p_prox = NULL;

  p_head = p;

  /* cria novos dados */

  strcpy (pessoa1.nome, "Pessoa2");

  pessoa1.sexo = 'M';

  pessoa1.idade = 18;

  pessoa1.p_prox = NULL;

  insere_pessoa (p_head, &pessoa1);

  p = p_head;

  while (p != NULL) {

    printf("%s %c %d\n", p->nome, p->sexo, p->idade);

    p = p->p_prox;

   }

  return 0;

}
```

Entrada

Não há entrada.

Saída

Este código só tem uma saída possível.

Samples Input	Samples Output
X	Pessoa1 F 42 Pessoa2 M 18

CUSTOM PROBLEM: 1032

LINGUAGEM: C++17

SOURCE CODE

```
#include <stdio.h>

int main() {

    /**
     * Escreva a sua solução aqui
     * Code your solution here
     * Escriba su solución aquí
     */

    return 0;
}
```

CONSTRUA A SUA SOLUÇÃO E ENVIE!

ENVIAR