

## ACH2023 ALGORITMOS E ESTRUTURAS DE DADOS I

Semestre 2020-2 - Exercício prático 1 – união de listas de inteiros – t.02

Estagiária PAE: Thais Souza (thais.donega@usp.br)

**Descrição do EP:** A partir do modelo disponibilizado no Tidia (*ep1-modelo.txt*), implementar as seguintes operações para união de duas listas de inteiros.

1. O objetivo do trabalho é implementar de forma correta e completa a função **união** utilizando apenas listas ligadas de implementação dinâmica como estrutura de armazenamento, e demais funções auxiliares para realização da operação solicitada.
2. As listas são de implementação dinâmica, encadeamento simples, sem nó cabeça, sem nó sentinela e sem circularidade. Não modifique estas definições, ou seu programa será invalidado.
3. A função recebe como entrada duas listas ligadas apontadas por p1 e p2 representando sequências **não decrescentes de inteiros, porém possivelmente repetidos**, e retorna uma lista ligada do mesmo tipo representando a fusão de p1 e p2 (ver exemplo) preservando a ordem dos elementos e **sem repetição**.

NO\* uniao(NO\* p1, NO\* p2)

4. EXEMPLO. Dadas as listas  $p1=\{1,2,2,6,8,9\}$  e  $p2=\{1,2,4,5,7,7\}$ , o resultado será  $\{1,2,4,5,6,7,8,9\}$ .
5. As listas p1 e p2 já serão fornecidas em ordem não-decrescente, mas podem ser vazias. A lista resultante deve conter todos os elementos de p1 e p2, sem repetição, e em ordem crescente.
6. RESTRIÇÕES DE IMPLEMENTAÇÃO:
  - a. Não **use nenhum vetor** na sua implementação. Se necessitar de estruturas auxiliares, use sempre listas ligadas de implementação dinâmica.
  - b. Não use variáveis globais. A função implementada deve definir localmente todas as variáveis e estruturas auxiliares, ou chamar funções auxiliares que o façam também em um escopo local.
7. Não exiba nenhuma mensagem na tela, nem solicite que o usuário pressione nenhuma tecla etc. Apenas implemente a função solicitada.
8. A função *main()* serve apenas para seus testes particulares, e não precisa ser entregue. Caso você prefira mantê-la no corpo do programa, pede-se apenas que *main()* seja a última função do programa, ou seja, que não haja nenhum código abaixo dela.
9. Seu programa será corrigido de forma *automática*, e por isso você não pode alterar as assinaturas da função solicitada, nem os tipos de dados ou especificações (*typedef*) do modelo fornecido.
10. O EP deve ser desenvolvido individualmente. Não tente emprestar sua implementação para outros colegas, em copiar deles, pois isso invalida o trabalho de **todos** os envolvidos.
11. O programa deve ser compilável no ambiente Windows com Codeblocks 13.12 ou superior. Será aplicado um desconto de até 30% na nota do EP caso ele não seja imediatamente compilável nesta configuração.
12. Programadores JAVA, cuidado: não existe inicialização automática de variáveis em C.

### O que/como entregar:

- A entrega será via upload no sistema e-disciplinas antes da data estipulada.
- Entregue apenas um arquivo texto com o código da função principal e funções auxiliares que ela invoca.
- A extensão do arquivo deve ser .cpp - **favor não compactar**.
- Preencha as funções nroUSP do código exemplo disponível para que você seja identificado.
- Na primeira linha do código, escreva um comentário "//" com seu nome.

### Prazos etc.:

O EP deve ser depositado no prazo definido na atividade cadastrada no sistema. Não serão aceitos EPs entregues depois do prazo, independentemente do motivo. Entregas no último dia são assim por conta e risco do aluno, e nenhum tipo de imprevisto de última hora (e.g., problemas de saúde, indisponibilidade de rede etc.) pode ser usado como justificativa para o atraso. O EP é uma atividade para ser desenvolvida ao longo de várias semanas, não nos últimos dias antes da entrega.

É responsabilidade do aluno que fez o *upload* do arquivo verificar se o mesmo foi corretamente recebido pelo sistema. Atrasos/falhas na submissão invalidam o trabalho realizado. Após o *upload*, verifique se você consegue abrir o arquivo depositado, e certifique-se de que é a versão correta do programa e que não está corrompido.

### Critérios de avaliação:

A função será testada com uma série de chamadas repetidas e consecutivas, com diversas listas de inteiros positivos como entrada. É assim importante assegurar que o seu programa funciona desta forma (por exemplo, chamando-o dentro de um laço *for*), e não apenas para um teste individual. Um teste é considerado correto se o resultado da soma for exatamente como o esperado, ou incorreto em caso contrário. Erros de alocação de memória ou compilação invalidam o teste, assim como a ausência de funções auxiliares necessárias para a execução do programa.

Este EP deve ser desenvolvido obrigatoriamente por *todos* os alunos de AED1. Sua nota é parte integrante da 1ª. avaliação e *não* é passível de substituição. Os EPs são o principal motivo de reprovação em ACH2023.