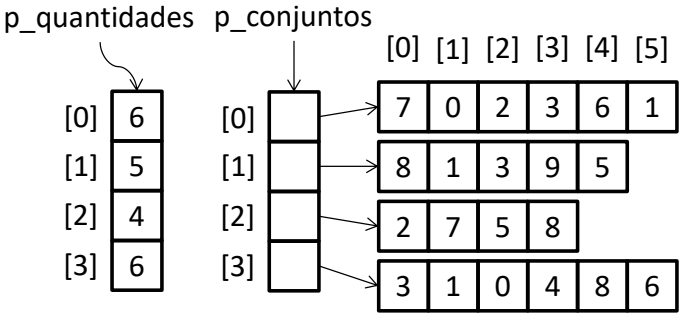
Universidade do Estado de Santa Catarina (UDESC)
Curso: Bacharelado em Ciência da Computação
Disciplina: Linguagem de Programação (LPG0001)
Prof. Rui Jorge Tramontin Jr.

Trabalho 2: Gerenciamento de Conjuntos

O objetivo do trabalho é implementar um programa que gerencie uma quantidade indeterminada de **conjuntos de valores inteiros**. O programa deve oferecer um *menu de texto* com as seguintes opções:

1. Criar um novo conjunto, inserindo dados nele;
2. Fazer a *união* entre dois conjuntos, gerando um novo conjunto;
3. Fazer a *intersecção* entre dois conjuntos, gerando um novo conjunto;
4. Fazer a *diferença* entre dois conjuntos, gerando um novo conjunto;
5. Mostrar todos os conjuntos;
6. Sair do programa.

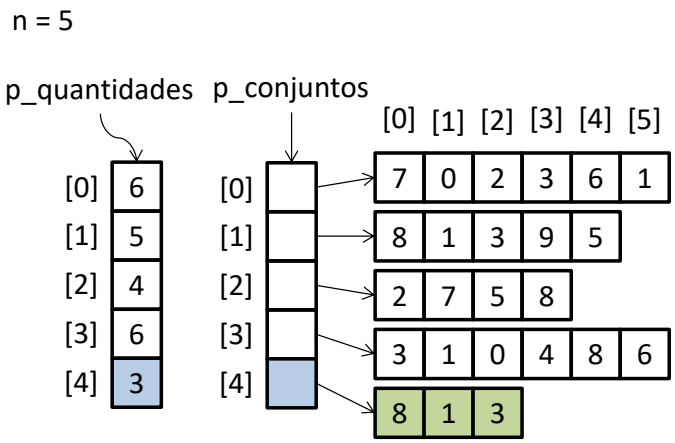
Os conjuntos são implementados por vetores alocados dinamicamente, cujos endereços são armazenados em um vetor de ponteiros, que também é realocado cada vez que o usuário criar um novo conjunto (opção 1) ou após uma operação de união (opção 2), intersecção (opção 3) ou diferença (opção 4). A figura a seguir ilustra o modelo proposto contendo 4 conjuntos:



A variável *n* (*int*) armazena a quantidade de conjuntos. O ponteiro *p_quantidades* (*int**) aponta para um vetor de inteiros que contém a quantidade de elelentos do conjunto na posição *i*. O ponteiro *p_conjuntos* (*int***) aponta para o vetor de ponteiros, onde cada ponteiro aponta para o conjunto na posição *i*.

Inicialmente, *n = 0* e os ponteiros apontam para *NULL*. Cada vez que um novo conjunto é criado (opções 1 a 4), *n* deve ser incrementado, e os ponteiros devem ser realocados em função de *n*. Em seguida, os dados devem ser armazenados conforme o modelo: o vetor de quantidades armazena a quantidade no índice *n-1*, e o vetor de conjuntos, no índice *n-1*, deve receber um *malloc()* em função da quantidade determinada. Por fim, os valores devem ser armazenados no vetor alocado.

Por exemplo, considerando os conjuntos apresentados, considere que o usuário deseja fazer a intersecção entre os conjuntos 1 e 3. A figura a seguir mostra como fica o modelo após esta operação (destaque em azul para as realocações e em verde para a alocação do novo conjunto):



Requisitos do programa (e algumas dicas)

- Para a criação de um conjunto, o programa deve solicitar quantidade de elementos (para fazer a alocação do vetor), e em seguida solicita os valores a serem inseridos no conjunto. **É preciso garantir que não haverá valores repetidos** (pois é um conjunto). A função de *busca*, feita em aula, pode ser útil para essa verificação;
- As funções de *união* e *intersecção* podem ser aproveitadas do conteúdo apresentado nas aulas e resolvido nos exercícios!
- A diferença entre os conjuntos A e B contém todos os valores que estejam em A, mas que não estejam em B. Portanto A - B não é igual a B - A. A implementação dessa operação em uma função (tal como feito na união e a intersecção) torna o código mais fácil de se reutilizar;
- A funcionalidade para mostrar todos os conjuntos pode ser facilmente implementada com a função para mostrar um vetor na tela;
- Caprichem nas mensagem para o usuário! Por exemplo, trate adequadamente quando não houver nenhum conjunto e o usuário escolher a opção para mostrar todos os conjuntos.
- Além disso, trate situações excepcionais, tais como quando o usuário escolher fazer a união incluindo um conjunto que não existe. Por exemplo, considerando o modelo apresentado na figura, não é possível fazer a união entre os conjuntos 2 e 6, pois não há o conjunto 6;
- Não se esqueça de desalocar toda a estrutura ao final do programa!

Critérios de avaliação

- Execução correta e alinhamento com o que foi solicitado neste enunciado.

Informações importantes:

- **Equipe:** 1 ou 2 alunos.
- **Entrega**: no Moodle, até o dia **02/04**.