O objetivo desse exercício é praticar questões de alocação de memória em classes e de polimorfismo, especialmente a sobrecarga de operadores.

Você deverá criar um novo tipo de dados chamado de HeapInt, que possui um funcionamento igual ao dos números inteiros mas que armazena o dado no heap. Ou seja, sua classe vai ter basicamente um apontador para um inteiro que deverá ser tratado adequadamente. Você deverá implementar os seguintes métodos e operadores:

- Um método construtor sem parâmetros e que inicializa o seu HeapInt com 0 (zero);

- Um método construtor que recebe um inteiro como parâmetro;

- Um método construtor que recebe um HeapInt como parâmetro (construtor de cópia);

- Um método destrutor;

- Sobrecarga do operador de atribuição (=) para fazer a atribuição de um inteiro para um HeapInt;

- Sobrecarga do operador de atribuição (=) para fazer a atribuição de um HeapInt para outro;

- Sobrecarga da operação de soma (+) para somar 2 números HeapInt;

- Sobrecarga da operação de subtração (-) para subtrair 2 números HeapInt;

- Sobrecarga da operação de igualdade (==) para comparar o valor de 2 números HeapInt;

- Sobrecarga dos operadores de entrada (>>) e saída (<<) para ler e imprimir um HeapInt.

Você deverá submeter o .hpp e o .cpp da sua classe. O arquivo main.cpp que utiliza a sua classe será fornecido.

Um exemplo de código segue abaixo:

```
  HeapInt a(10);
  HeapInt b;

  cin >> b;
  if (a==b)
    cout << "Bingo!" << endl;

  HeapInt c = a;
  HeapInt d;
  HeapInt e;

  d = 5;
  e = a + b + c + d;
  cout << a << " " << b << " " << c << " " << d << " " << e << endl;
```

Para a entrada 10, o valor impresso seria:

```
Bingo!
10 10 10 5 35
```