	Universidade Federal de Campina Grande **Departamento de Sistemas e Computação** **Disciplina: *Técnicas de Programação*** **PROVA DO MÓDULO I – PRÁTICA – TURMA: 01** **SEMESTRE 2015.1 – DATA: 26/05/2015**	
Matrícula	**Nome**	**Nota**

1. Construir um programa C++ que contenha uma *struct* e uma *enum* para armazenar inteiros, na qual os dois primeiros membros correspondam a dois números quaisquer, enquanto os demais correspondam a resultados de seguintes operações *bit* a *bit*: (i) *NÃO* (*NOT*); (ii) *E* (*AND*); (iii) *OU* (*OR*); (iv) *OU EXCLUSIVO* (*EXCLUSIVE OR*); (v) *DESLOCAMENTO PARA A ESQUERDA* (*LEFT SHIFT*); e (vi) *DESLOCAMENTO PARA A DIREITA* (*RIGHT SHIFT*). Os resultados de tais operações deverão ser inicializados por funções que recebam a referência à estrutura e inicializem seus membros. **1,5**

   Detalhes de implementação:

   - Deverão ser criadas as funções *not(struct bitwise *est)*, *and(struct bitwise *est)*, *or(struct bitwise *est), xor(struct bitwise *est)*, *lefts(struct bitwise *est)* e *rights(struct bitwise *est)*;
   - As operações bit-a-bit deverão ser testadas a partir de um programa que solicite do usuário os operandos e a operação desejada; e
   - O usuário será capaz de solicitar tantas operações com pares de operandos quantas queira e de encerrar o programa de teste quando lhe for conveniente.

2. Com base na seguinte interface de uma classe C++ denominada *Passagem*, implementar: **1,5**

```
class Passagem{
   string local; // Localidade de destino da viagem
   double preco; // Preco da passagem
   string nome; // Nome do vendedor da passagem
 public:
   Passagem(string local, double preco, string nome);
   string Exibe_local( ) const;
   double Exibe_preco( ) const;
   string Exibe_nome( ) const;
   void Altera_local(string novo_nome);
   void Altera_preco(double novo_preco);
   void Altera_nome(string novo_nome);
   void Exibe_tudo( ) const; // Exibe o nome do vendedor, a localidade
                             // e o preco da passagem
   double Exibe_vendas(int num_pass) const; // Exibe o valor total de passagens
                                            // vendidas, a partir do numero de
                                            // total número total de passagens
   bool operator< (Passagem pass2); // Compara precos de passagens
}
```

i. todas as funções-membros da classe;
ii. um programa de teste, considerando-se os seguintes dados:

Localidades e preços (partida de Campina Grande)

Localidade	Preço (R$)
Queimadas	**3,35**
Fagundes	**4,10**
Monteiro	**33,00**
Ouro Velho	**34,00**
João Pessoa	**51,35**
Cajazeiras	**75,60**

Vendedores: **Margareth**/ **Kaio**/ **Marina**/ **Ravi**/ **Giuseppe**/ **Wesley**

3. Construir uma classe C++ *Data*, para a qual devem ser implementados: **2,0**

a) C*onstrutores*, cujos argumentos podem ser *nenhum*, *um* (dia, mês ou ano), *dois* (dia e mês, dia e ano ou mês e ano) e *três* (dia, mês e ano). Se chamado sem argumentos, o construtor deve configurar os valores de dia, mês e ano para a data de *hoje*. Se chamado com 2 argumentos (dia e mês), o ano é configurado para a data de *hoje*;

b) Funções-membros para *retornar dia, mês e ano*, *ajustar a data para um dado valor <u>especificado pelo usuário</u>* e *verificar se uma data qualquer <u>é a data atual</u>*;

c) Duas funções-membros de sua escolha que sejam úteis para a manipulação de **Data** pelo usuário; e

d) Um programa de teste para a implementação.

ÊXITO!