**Projeto 2:** identificação das rotas de menor custo em uma cidade.

**Aluno:** Paulo Victor de Aguiar

TEG 2015/1

**Introdução:**

O projeto tem como objetivo determinar a menor distancia e tempo de uma cidade representada por um grafo de um ponto para o outro. Para execução foi criada uma matriz IJ onde a coordenada guarda os valores de tempo e distancia da linha I para a coluna J e para calcular a menor distancia e tempo foi utilizado o algoritmo de Floyd-Wharshall para poder trabalhar com várias distancias ao mesmo tempo. As linhas e colunas representam as esquinas (vértices) da cidade. O código foi divido em três partes: Descritor.h, Floyd.c e Programa.c.

**Descritor.h**

A parte inicial do código e a mais simples. Declara o typedef struct Rua para armazenar os valores int tempo e int distancia que serão utilizados na matriz.

**Floyd.c**

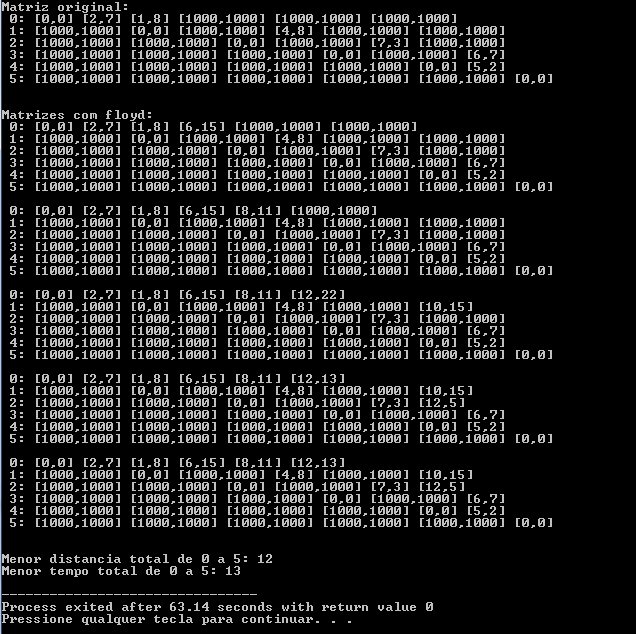
Contém o algoritmo de Floyd-Wharshall para calcular a menor distancia e tempo e recebe como parâmetros uma matriz bidimensional e o numero de vértices. A primeira parte do código é criar uma matriz tridimensional KIJ e é alocada conforme o numero de vértices, ou seja, é um conjunto de matrizes bidimensionais do tamanho do numero de vértices. A primeira matriz desse conjunto é criada em base da original com a aplicação do algoritmo de Floyd-Wharshall. Com a primeira criada, o processo se repete com as demais e por fim a matriz original é atualizada com a última matriz gerada pelo algoritmo.

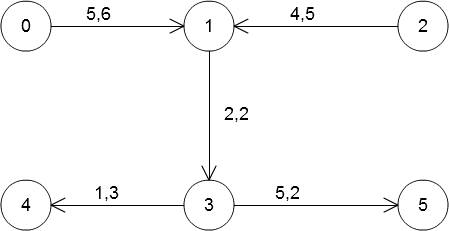
**Programa.c**

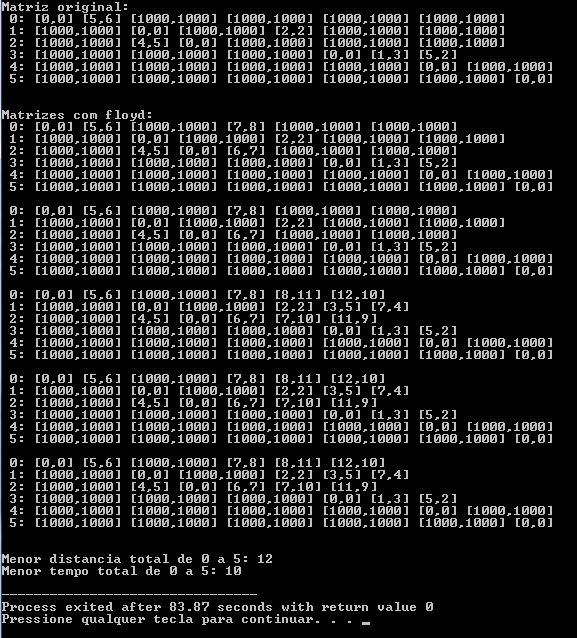
Possui a função principal que executa o programa. Inicialmente cria uma matriz do tipo Rua alocada conforme o numero de esquinas recebidas e é inicializada com o tempo e distancia igual a 1000 (infinito). Em seguida recebe a esquina inicial, a esquina final e o numero de ruas. Com base do numero de ruas são feitas linhas de leitura para receber a esquinas e formam a rua e o tempo e distancia entre elas. Por fim imprime a matriz do grafo gerado e as matrizes do algoritmo de Floyd-Wharshall, no final mostra o menor tempo e distancia total em base da ultima matriz gerada.

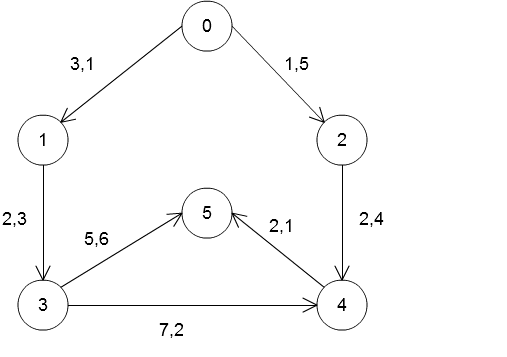
**Grafos:**

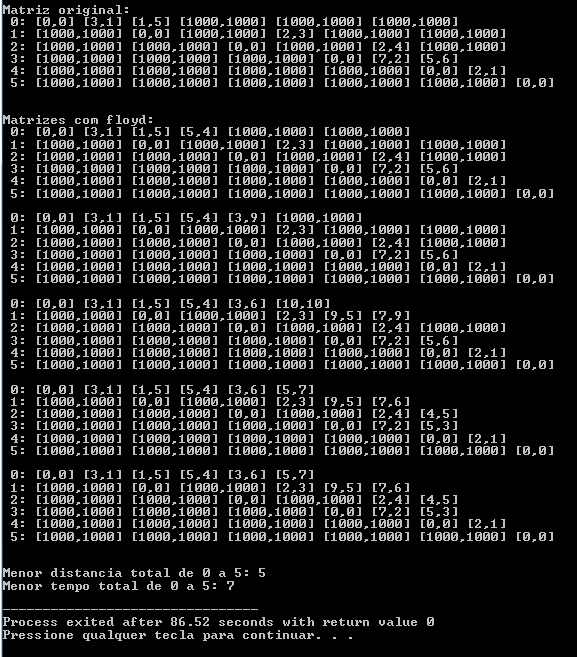
****

****

****

****

****

****