

**Universidade Federal do Espírito Santo - CT - DI**  
**Trabalho Computacional - PD II - Engenharia Elétrica**  
**Profa. Maria Cristina Rangel**

Considere o esboço do estado do ES e  $n = 30$  cidades enumeradas de  $1, 2, \dots, n$ . Os nomes das cidades e as suas respectivas coordenadas  $x$  e  $y$  estão armazenadas no arquivo **nome-coord.txt**. Cada estrada entre duas cidades possui um custo por quilômetro que deve ser multiplicado pela distância para calcular o custo de viagem entre as cidades. Além disso, o hotel em cada cidade possui um determinado valor para a diária. Os valores das diárias e os custos por quilômetro estão armazenados no arquivo **diaria-custo.txt**.

Faça um programa na linguagem C para:

1. Ler o arquivo de dados **nome-coord.txt**
2. Ler o arquivo de dados **diaria-custo.txt**
3. Construir uma matriz  $D_{n \times n}$  de distâncias entre todas as cidades e armazenar em um arquivo que se chamará **distancia.txt**
4. Construir uma matriz  $C_{n \times n}$  de custos de viagem entre todas as cidades e armazenar em um arquivo que se chamará **custo.txt**
5. Imprimir na tela a(s) cidade(s) mais ao norte, mais ao sul, mais a oeste, mais a leste e a mais central
6. Construir (imprimir na tela) o caminho da **cidade 1** até a **cidade  $n$** , passando por todas as cidades na ordem  $1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow 29 \rightarrow n$ . Calcular (imprimir na tela) a distância total deste caminho (usando a matriz  $D$ ) e o custo total desta viagem (usando a matriz  $C$ ), considerando que o viajante dormirá uma noite em cada cidade, exceto a cidade 1.
7. Construir (imprimir na tela) um caminho da **cidade 1** até a **cidade  $n$**  seguindo a lei de formação: a cidade seguinte no caminho é a mais próxima e sempre considerando as cidades a frente na ordem fornecida no arquivo **nome-coord.txt** e ilustradas no gráfico. Exemplo: se estamos na **cidade 10** a cidade seguinte será a **cidade 15** e não a **cidade 8 ou 9**. Se houver empate com relação à distância, escolher a cidade que diminui o número total de cidades visitadas no caminho (a cidade

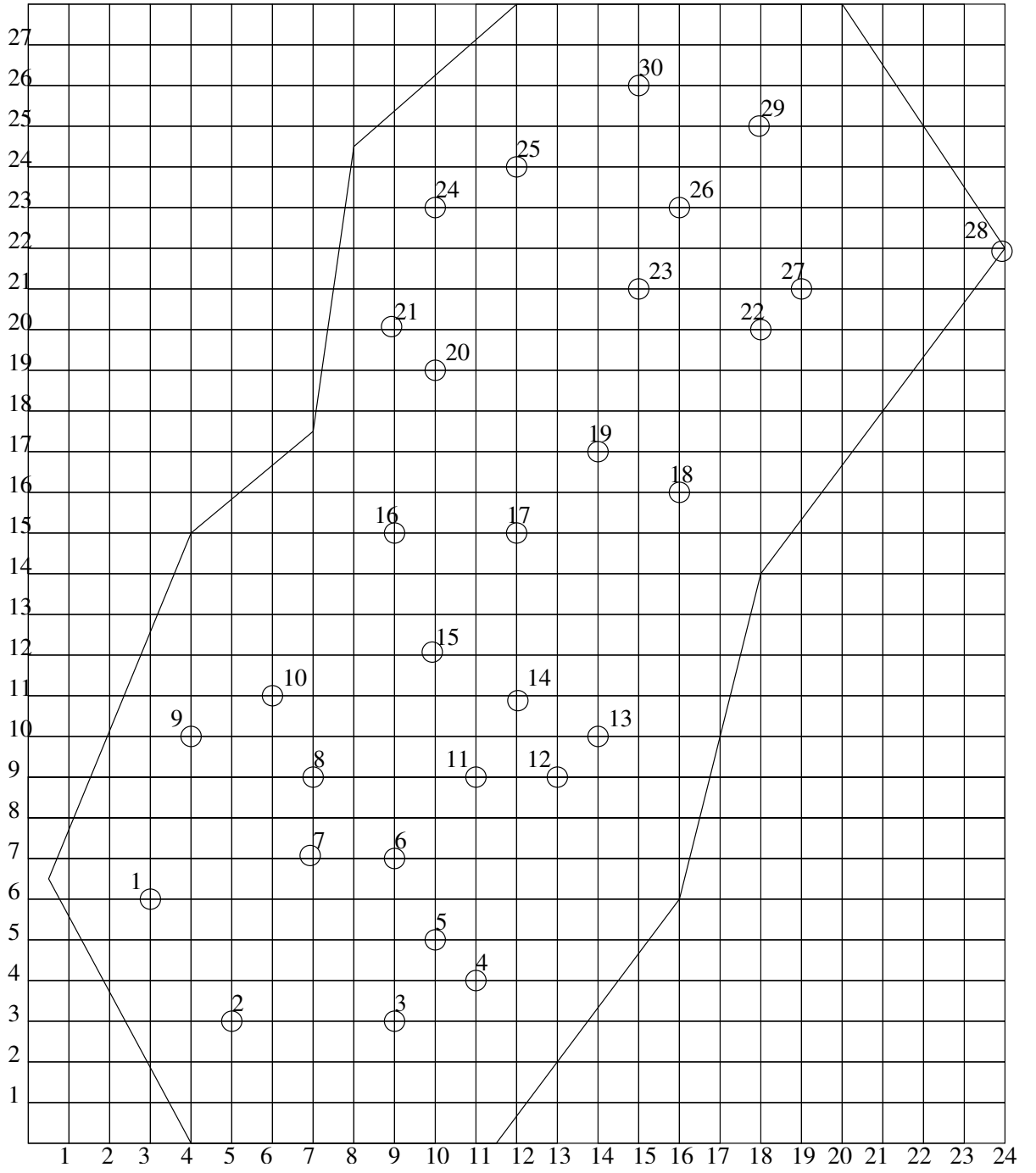
de maior ordem). Calcular (imprimir na tela) a distância total deste caminho e o custo total desta viagem.

8. Construir (imprimir na tela) um caminho da **cidade 1** até a **cidade  $n$**  seguindo a lei de formação: a cidade seguinte no caminho é a menos custosa e sempre considerando as cidades a frente na ordem fornecida no arquivo **nome-coord.txt** e ilustradas no gráfico. Se houver empate com relação ao custo, escolher a cidade que diminui o número total de cidades visitadas no caminho (a cidade de maior ordem). Calcular (imprimir na tela) a distância total deste caminho e o custo total desta viagem.

#### Observações importantes:

- Para calcular o custo de viagem entre as cidades  $i$  e  $j$ , utilize a expressão:  $CV_{ij} = c_{ij} \times d_{ij} + diaria_j$
- Os arquivos de dados **nome-coord.txt** e **diaria-custo.txt** estarão disponíveis na página da disciplina ([www.inf.ufes.br/~crangel/pd2-ele-082.html](http://www.inf.ufes.br/~crangel/pd2-ele-082.html))
- **Data de Entrega:** O trabalho deverá ser entregue até às 23:59 horas do dia 28/11/2008 (sexta-feira).

A seguir, o esboço do ES com as  $n = 30$  cidades.



### **Forma de Entrega:**

1. Compacte o arquivo texto (o nome do arquivo DEVE ser caminho.c) com o código fonte do programa do seu trabalho e envie o arquivo compactado para o e-mail **crangel@inf.ufes.br**. O nome do arquivo compactado deverá ser trab.zip (por favor, não enviem .rar)
2. O assunto do e-mail deverá ser o seguinte (somente o que está entre aspas duplas): "pd2:trab:nome1:nome2:nome3", onde nome1, nome2 e nome3 (máximo 3 componentes) são os nomes dos integrantes do grupo. Substitua nome1 (e demais) pelo primeiro nome e último sobrenome, separados por espaços. Compacte o arquivo fonte utilizando o programa (ou comando) zip e envie o arquivo compactado em anexo. Ressaltamos que o arquivo trab.zip gerado deverá conter apenas o arquivo com o código fonte do seu trabalho (não podem ter arquivos executáveis ou qualquer outro arquivo)
3. O recebimento dos trabalhos é automatizado. Siga as instruções à risca pois algum erro na submissão pode inviabilizar a entrega do seu trabalho. Não deixe para enviar seu trabalho nos momentos finais de seu prazo. É comum a ocorrência de problemas em virtude de erros na submissão. Logo, enviem com algumas horas de antecedência para que haja tempo hábil para eventuais correções

Veja abaixo um exemplo de um e-mail de envio do trabalho do grupo formado por João da Silva e José Geraldo Castro (enviado por João da Silva). Apenas um integrante do grupo envia o trabalho. Não use acentos, cedilhas ou qualquer outro caractere especial.

**Para:** crangel@inf.ufes.br

**De:** Joao da Silva

**Assunto:** pd2:trab:Joao Silva:Jose Castro

**Anexo:** trab.zip

### **Atenção:**

1. No assunto, a disciplina (pd2) e a identificação do trabalho (trab) devem ser escritos todos em letras minúsculas
2. NÃO escreva o seu nome com caracteres estendidos (ã, ç, é, etc)

3. Apenas um mail por trabalho deve ser enviado

**Outras Observações Importantes:**

1. Os trabalhos serão verificados automaticamente por uma ferramenta de detecção de plágio. Em caso de detecção de cópia (parcial ou integral), todos os envolvidos recebem nota ZERO. Em outras palavras, tanto os alunos que copiaram quanto os que deixaram copiar recebem ZERO
2. Enviem o trabalho no prazo especificado e no formato especificado. Trabalhos recebidos fora do prazo ou em formato inadequado recebem nota ZERO
3. O trabalho deve ser enviado estritamente para o e-mail especificado acima
4. Trabalho que não compila recebe nota ZERO. Não adianta submeter
5. Os trabalhos serão compilados e verificados usando o compilador gcc no sistema operacional Linux
6. Os programas serão avaliados pela sua correção durante a execução e também pelo estilo de programação. Serão observados particularmente se os programas possuem os comentários apropriados, se usam nomes significativos para as variáveis e funções, se o código está indentado corretamente e se utilizam modularização sempre que possível e apropriado