Rudini Menezes Sampaio
Aluno do Instituto Tecnológico de Aeronáutica (ITA)
Bolsa PIBIC/CNPq
Orientador: Horácio Hideki Yanasse
Laboratório Associado de Computação e Matemática Aplicada (LAC)
Instituto Nacional de Pesquisas Espaciais (INPE)
Avenida dos Astronautas, 1758 - Caixa Postal 515

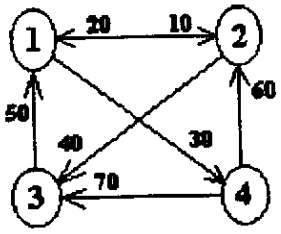
Problemas urbanos práticos tais como entrega de correspondência, limpeza de rua, coleta de lixo, serviços de ônibus para melhor atendimento à população, reparos de cabines telefônicas, remoção de neve e problemas de seleção de lugares para localização de serviços de emergência (pronto socorros, bombeiros,...) ou não emergência (terminais de transporte e etc...).podem ser modelados como problemas em grafos.

A teoria de grafos direcionada para a solução de tais problemas possui várias vantagens entre elas a fácil descrição do problema para parâmetros da teoria e a facilidade com que se pode encontrar soluções aproximadas para problemas matematicamente complexos.

O rápido avanço na tecnologia dos computadores proporcionou uma atenção especial para esta teoria, ocasionando o desenvolvimento de vários algoritmos que solucionem estes problemas. Tais algoritmos envolvem problemas de menor caminho, árvores de custo mínimo, tour euleriano e caminho euleriano, carteiro chinês, caixeiro viajante e outros.

Sendo assim, este trabalho tem por objetivo a implementação de tais algoritmos, criando um ambiente em que se possa manipular grafos e onde seja possível a visualização de tais grafos e dos resultados da aplicação dos algoritmos sobre eles.

Para isto, foi preciso definir uma estrutura de dados para a entrada de grafos, o que foi feito a partir de arquivos textos com um cabeçalho nas primeiras linhas para identificação interna e, nas restantes, cada linha representa um nó informando os nós a quem este se liga seguido do valor do arco correspondente. Para exemplificar, o grafo da figura 1, vem seguido do respectivo arquivo texto que o representa. A forma de desenho do grafo na tela é resolvido durante o desenvolvimento do programa e está bem explicado no relatório final, bem como no "Help" do programa executável, inclusive com as melhorias de tal estrutura.



Esboço do arquivo de representação:

2 10 4 30 *0*	(Ligações do Nó 1)
1 20 3 40 *0*	(Ligações do Nó 2)
1 50 *0*	(Ligações do Nó 3)
2 60 3 70 *0*	(Ligações do Nó 4)

Figura 1: Grafo seguido do esboço do arquivo que o representa, para exemplificar a estrutura de dados utilizada.

Bolsa PIBIC/CNPq

INPE/LAC

Estudo e Implementação de Algoritmos para Resolução de Problemas Combinatórios em Grafos

Bolsista: Rudini Menezes Sampaio
Orientador: Horácic Hideki Yanasse

Junho de 1997

INTRODUÇÃO

Problemas urbanos práticos tais como entrega de correspondência, limpeza de ruas, serviços de ônibus para melhor atendimento à população, reparos de cabines telefônicas, remoção de neve e problemas de seleção de lugares para localização de serviços de emergência (pronto socorros, bombeiros, ...) ou não emergência (terminais de transporte, ...) podem ser modelados como problemas em grafos.

A teoria de grafos direcionada para a solução de tais problemas possui várias vantagens entre elas a fácil descrição do problema para parâmetros da teoria e a facilidade com que se pode encontrar soluções aproximadas para problemas matematicamente complexos.

O rápido avanço na tecnologia de computadores proporcionou um atenção especial para esta teoria, ocasionando o desenvolvimento de vários algoritmos que solucionem estes problemas. Tais algoritmos envolvem problemas de menor caminho, árvores de custo mínimo, tour euleriano e caminho euleriano, carteiro chinês, caixeiro viajante e outros...

Com base nestes argumentos, desenvolveu-se um projeto para elaborar um software capaz de manipular grafos em geral, sendo possível carregar grafos, criar novos grafos, modificar seus parâmetros e salvar modificações feitas, bem como utilizar os algoritmos acima citados. Este software deveria ter uma boa ambientação e ser de fácil entendimento e uso (user friendly), deveria estar bem modularizado para ser possível a reutilização de partes do programa caso necessário e não ocasionar erros visto que um erro interno ou de informação equivocada pode gerar enormes problemas.

Os conceitos acima citados descrevem um programa que deva ter robustez, reusabilidade e manutenibilidade, conceitos estes que são palavras-chave em processos de engenharia de software.

ATIVIDADES REALIZADAS

Para poder-se desenvolver tal software, escolheu-se a linguagem Pascal, devido ao fato de ser a linguagem de maior conhecimento do bolsista. Desenvolveu-se então o programa "Grafos.pas" e para isto foi necessário um conhecimento aprofundado da linguagem para as situações abaixo:

1. Divisão do programa em unidades (arquivos externos).
2. Utilização da unidade gráfica do Pascal.
3. Manipulação de áreas de memória para armazenamento de telas gráficas.
4. Utilização de funções referentes ao Mouse.

O item **1** acima foi importante para a modularização do programa tendo em vista o fácil entendimento do mesmo e a sua fácil manutenção. Dividiu-se então o programa em oito(8) unidades de rotinas e um programa principal que as utiliza. O programa principal denomina-se "Grafos" e as unidades são: File_, Option, Jan345, Desenha, Implemen, Algorit, Mouse e Geral.

O item **2** acima foi importante para ser possível o desenho de janelas, menus, botões, textos em tela gráfica, desenho do grafo com mudança de tipo de traço e cores, e etc... como pode-se ver na execução do programa principal.

O item **3** acima foi importante para ser possível o aparecimento de janelas e mensagem sobre desenhos sem apagá-los, ou seja, gravou-se na memória a tela antes do aparecimento da janela, mostrou-se a janela e, no término de sua função, recuperou-se a tela gravada anteriormente. Esta função é exercida pela rotina Grava_Cola(x) da unidade Geral. Se x=10, então o programa grava na memória a tela atual. Se x=01, então o programa "Cola" na tela a figura que está em determinada posição da memória.

O item **4** acima foi importante para ser possível a escolha fácil de opções apenas clicando com o botão esquerdo em círculos ou botões do programa, a saída de janelas e menus apenas clicando com o botão direito, a limitação da região da tela gráfica de posicionamento da seta do mouse., a escolha da posição e a habilitação ou não do aparecimento da seta do mouse. Foi necessário um estudo de Assembler, principalmente das interrupções do DOS que manipulam o mouse (INT 33), para desenvolvimento e aperfeiçoamento de tais rotinas no Pascal. Essas funções estão descritas na unidade Mouse.

Para desenvolver o software, foi necessário estudar a teoria de Grafos. Como implementá-los, como utilizá-los para diversas situações e como manipulá-los para execução de algoritmos. A forma de entrada dos grafos para o programa será descrita mais abaixo.

Precisou-se ter conhecimento de estrutura de dados para poder armazenar grafos, em forma de matriz de adjacências e lista de adjacências. Implementamos as duas formas para verificar qual a que possui melhor desempenho, ocupa menos espaço na memória e é de fácil implementação.

Verificamos que a lista de adjacências ocupa menos espaço na memória, mas, mesmo assim, escolheu-se a matriz de adjacências devido a facilidade de implementação, de entendimento e de uso para algoritmos. A estrutura em forma de lista de adjacências é mostrada na unidade Desenha em forma de comentário, para possível uso posterior. A implementação em forma de matriz de adjacências é implementada na unidade Desenha, onde se tem uma Matriz NxN (onde N é o número de nós do grafo) de inteiros que representam os valores dos arcos. A_{ij} é igual ao valor do arco de ligação entre os nós I e J.

Se não há arco de I para J, então $A_{ij}=\infty$ (infinito; na verdade, é um valor muito alto que se vê como se fosse infinito). Implementa-se infinito com o valor GIG do programa que corresponde ao máximo valor inteiro.

Para implementação de algoritmos, precisou-se estudar na bibliografia vários conceitos da teoria de grafos. Precisou-se ter o vetor dos graus de cada de nó (número de nós a quem se liga), verificar se um dado grafo era direcionado ou não, se era conexo ou não, se era fortemente conexo ou não, se possui ou não tour euleriano ou caminho euleriano, e etc.. Várias algoritmos necessitam dessas informações para serem executados.

Observou-se tais algoritmos e procurou-se otimizá-los observando seu desempenho. Para isso foram necessários muitos testes, tanto de funcionamento quanto de visualização (interface).

As rotinas estão bem comentadas e bem modularizadas podendo ser entendida por todos que se interessarem. As unidades desenvolvidas podem também ser utilizadas por outros programas, observando-se apenas as entradas necessárias para cada uma delas e as saídas que estas fornecem.

ENTRADA DE DADOS

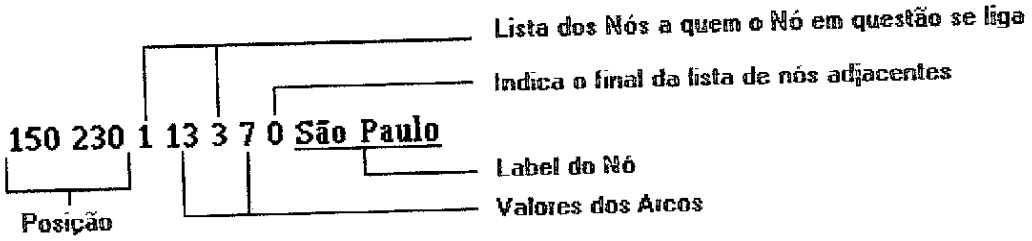
Para poder manipular grafos, foi preciso definir uma entrada de dados adequada para representar grafos e que fosse de fácil entendimento. Isso foi feito utilizando-se arquivos texto, onde tem-se nas primeiras linhas um cabeçalho para identificação interna e, nas outras, cada linha representa um nó do grafo informando os nós a quem se liga, o valor dos arcos e o "Label" do nó (Nome dele).

Existem três tipos de arquivos que podem representar grafos: Tipo 1, 2 e 3.

Exemplos:

Arquivo do Tipo 1	**Arquivo do Tipo 2**	**Arquivo do Tipo 3**
TIPO1 3 12 4 17 0 Rio 1 13 3 7 0 São Paulo 4 29 2 10 0 Minas 1 18 0 Fortaleza	TIPO2 100 100 3 12 4 17 0 Rio 150 230 1 13 3 7 0 São Paulo 200 170 4 29 2 10 0 Minas 180 100 1 18 0 Fortaleza	TIPO3 100 100 3 4 0 Rio 150 230 1 3 0 São Paulo 200 170 4 2 0 Minas 180 100 1 0 Fortaleza

A figura abaixo mostra cada item em uma linha para um grafo do tipo 2:



No tipo 1, pode-se omitir a posição na tela do nó representado pela linha. Nesse caso, o programa escolhe uma posição para os nós na tela, onde os nós ficam sobre uma circunferência com centro no centro da tela e diâmetro igual a altura da tela.

No tipo 2, não se pode omitir nada. É necessário informar a posição do nó na tela, bem como aqueles a quem ele se liga seguido do valor do arco.

No tipo 3, pode-se omitir o valor dos arcos, sendo necessário apenas a lista dos nós a quem o nó em questão se liga. O valor do arco é então igual a distância entre os nós citados. Esse caso é importante quando se trata de grafos que representam localidades como cidades, rios e etc...

O formato do arquivo texto que contém o grafo deve indicar seu tipo na primeira linha com as palavras TIPO1 ou TIPO2 ou TIPO3, indicando qual o tipo do grafo. As próximas linhas a partir da segunda representam os nós. Caso o grafo seja do tipo 2 ou 3, então em cada linha deve haver dois números iniciais que indicam a posição X e Y do nó na tela.

Em seguida, deve constar em cada linha a lista dos nós a quem o nó em questão se liga, seguido do respectivo valor do arco caso seja do tipo 1 ou 2. No final de cada lista desta, deve haver o número zero(0). Após este número, escreve-se então o Label do nó que seria um nome para referência. O Label pode ser um nome qualquer com menos de 256 caracteres. Pelo tamanho do arquivo, o programa sabe o número de nós do grafo.

Especificada a entrada de dados, o programa identifica o tipo do grafo e o desenha conforme tal.

A partir de tal entrada e identificado o seu tipo, o programa o armazena em uma matriz de adjacências podendo ser executado algoritmos e outras rotinas de manipulação. Qualquer mudança realizada é gravada em um arquivo do programa, esperando que o usuário a salve ou não. Ao término do programa, este manda uma mensagem perguntando se o usuário deseja salvar as modificações feitas.

FUNCIONAMENTO DO PROGRAMA

Desenvolveu-se 5 menus: FILE, OPTIONS, HELP, ABOUT e EXIT. O funcionamento destes menus são descritos nas unidades File_, Option e Jan345.

O menu FILE (Ilustrado em fig. 1) permite:

- Criar um grafo novo entrando o nome do arquivo, o tipo e a posição do primeiro nó caso seja do tipo 2 ou 3. (NEW - Ilustrado em fig. 1a)
- Carregar um grafo entrando o nome do arquivo. O programa então desenha o grafo na tela e executa a opção que estiver indicada em Opções para Execução no menu OPTION: Labels, Características ou Algoritmos. (LOAD- ilustrado em fig. 1b)
- Salvar o grafo, caso alguma modificação tenha sido feita. (SAVE)
- Gravar o grafo em outro arquivo, entrando para isso o nome do outro arquivo. (SAVE AS - Ilustrado em fig. 1b)
- Modificar parâmetros do Grafo (EDIT - Ilustrado em fig. 1c) , onde se têm as seguintes opções:
  - ⇒ Incluir Nó
  - ⇒ Excluir Nó
  - ⇒ Incluir Arco (Ilustrado em fig. 1d)
  - ⇒ Excluir Nó
  - ⇒ Mudar Labels

O menu OPTION (Ilustrado em fig. 2) permite escolher as seguintes opções:

- Opções Gerais:
  - ⇒ Visualizar Grafo
  - ⇒ Posição do Mouse
  - ⇒ Algoritmo com Delay (Algoritmo executado passo a passo)
- Opções de Visualização (escolher o que deve aparecer ou não no desenho do grafo) :
  - ⇒ Valores dos Arcos
  - ⇒ Círculos
  - ⇒ Setas
  - ⇒ Arcos
  - ⇒ Estrada (Os arcos em forma de estrada, caso seja do tipo 3)
- Opções de Execução (escolher aquilo que deve ser mostrado ou executado após o desenho do grafo ou clicando em "Aplicar"):
  - ⇒ Labels do Grafo
  - ⇒ Características do Grafo
  - ⇒ Algoritmos
    - ⇒ Menor Caminho
      - ⇒ Dijkstra
      - ⇒ Floyd
    - ⇒ Minimum Spanning Tree (Árvore de cobrimento mínimo)

O que for escolhido no menu OPTION em Opções de Execução será executado, toda vez que o grafo for redesenhado ou quando clicar-se em "Aplicar". "Labels do Grafo" mostra os nomes dados aos nós no arquivo, podendo ser modificados em Edit do menu FILE. "Características do Grafo" contém uma lista informando suas peculiaridades, como se é direcionado ou não, se é conexo ou não, e etc..."Algoritmos" permite a execução de algum algoritmo. Tais algoritmos são explicados posteriormente.

O menu HELP auxilia o usuário a compreender algumas questões básicas do funcionamento do programa como a utilização do mesmo e a entrada de dados dos arquivos que contém grafos. Está ilustrado nas figuras 3a e 3b.

O menu ABOUT informa quem são os desenvolvedores do software, bolsista e orientador, bem como o tipo da bolsa, órgão responsável e local de trabalho. Está ilustrado na figura 4.

O menu EXIT envia uma mensagem perguntando se usuário quer mesmo sair do programa. Caso queira, o programa fecha a tela gráfica, salva o arquivo utilizado e finaliza o programa. Está ilustrado na figura 5.

FUNCIONAMENTO DOS ALGORITMOS

Desenvolveu-se 6 algoritmos de manipulação de grafos:

1) Dijkstra (Menores caminhos de um dado nó para os outros nós).
2) Floyd (Menores caminhos entre todos os pares de nós).
3) Prim (Árvore de cobrimento mínimo).
4) Conexo (Verificar se é ou não conexo ou fortemente conexo).
5) Tour Euleriano (Verificar se possui tour euleriano).
6) Caminho Euleriano (Verificar se possui caminho euleriano).

Esses algoritmos estão descritos na sua forma pura na unidade "Algorit" e descritos na sua forma de implementação para o programa na unidade "Implemen".

Explicações sobre os algoritmos acima:

1) Dijkstra:

O algoritmo de Dijkstra necessita de um nó como entrada, visto que ele fornece o vetor dos valores dos menores caminhos entre esse dado nó e os outros nós e o vetor que possibilita marcar os menores caminhos.

Sua execução se encontra nas figuras 7a, 7b, 7c, 7d e 7e abaixo, utilizando como entrada o grafo da figura 6.

2) Floyd:

O algoritmo de Floyd fornece a matriz dos menores caminhos entre todos os pares de nós e a matriz que marca o percurso do menor caminho entre os nós do par.

Sua execução se encontra na figura 8, utilizando como entrada o grafo da figura 6, e como saída uma matriz que é mostrada na tela. Caso o número de nós do grafo seja maior que 10, é possível visualizar toda a matriz movendo-a com as setas do teclado.

3) Prim:

O algoritmo de Prim fornece a árvore do grafo cujo custo total (Soma de seus arcos) é mínimo, ou seja, a árvore de cobrimento (ou custo) mínimo. Esse algoritmo é utilizado apenas para grafos não direcionados e conexos. Mas uma nova implementação foi feita permitindo que para um grafo desconexo (que representa vários grafos conexos) retorne as árvores de custo mínimo para cada um de seus subgrafos conexos. Exemplos da execução deste algoritmo se encontram nas figuras 9 e 10, para um grafo conexo e para um grafo desconexo respectivamente.

4) Conexo:

Verifica em que nível de conexão está um dado grafo, podendo ter 3 níveis: Desconexo, Simplesmente conexo e Fortemente conexo.

Para verificar os níveis de conexão, analisa-se a matriz de menores caminhos de Floyd. Para ser Fortemente conexo, todos os valores devem ser diferente de infinito (∞). A verificação para o caso de ser Simplesmente conexo é ligeiramente mais complexa estando descrita no programa fonte. Caso não seja nem Fortemente conexo nem Simplesmente conexo, então o grafo é desconexo.

Nas figuras 11, temos exemplos da execução deste algoritmo.

5) Tour Euleriano:

Verifica se dado grafo possui ou não Tour Euleriano. Tal verificação já é feita assim que o grafo é carregado, na unidade Desenha.

Nas figuras 11, temos exemplos da execução deste algoritmo.

6) Caminho Euleriano:

Verifica se dado grafo possui ou não Caminho Euleriano. Tal verificação também já é feita assim que o grafo é carregado, na unidade Desenha.

Nas figuras 11, temos exemplos da execução deste algoritmo.

BIBLIOGRAFIA

•Aho, A.V.; Hopcroft, J.E.; Ullman, J.D., **The design and analysis of computer algorithms**, Addison-Wesley, 1974.

•Boaventura Netto, P.O., **Grafos: teoria, modelos e algoritmos**, Edgard Blücher, 1996.

•Campello, R.E.; Maculan, N., **Algoritmos e heurísticas**, EDUFF, 1994.

•Duncan, R., **Advanced MSDOS programming,** Microsoft Press, 1986.

•Larson, R.C.; Odoni, A.R., **Urban operations research**, Prentice Hall, 1981.

•Mokarzel, F.C., Apostila do curso Estrutura de dados, Computação, ITA, 1990.

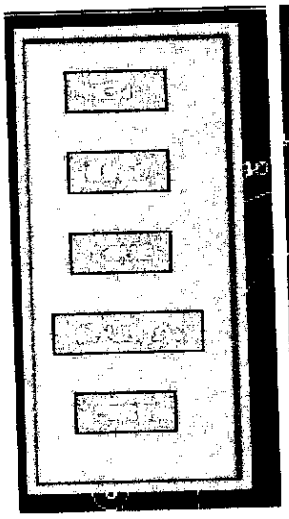


Figura 1: Menu FILE

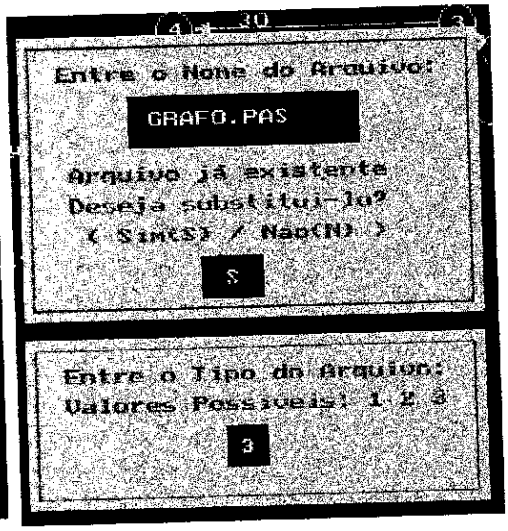


Figura 1a: Execução de NEW

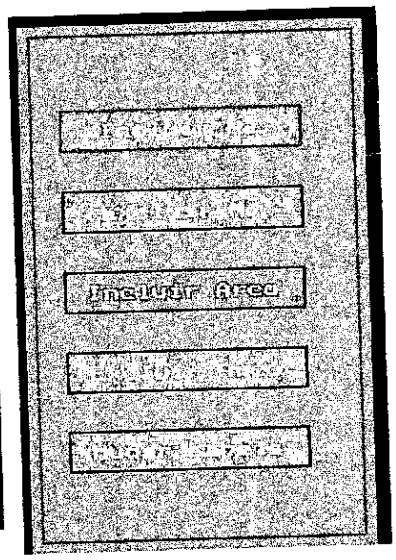


Figura 1c: Execução de EDIT

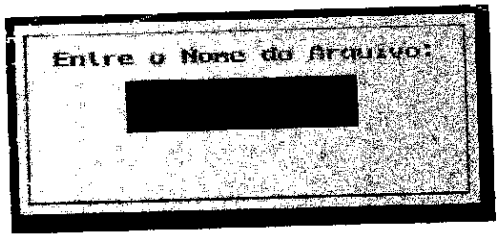


Figura 1b: Execução de LOAD e SAVE AS

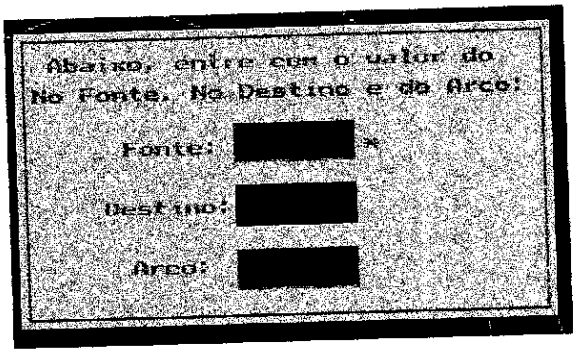


Figura 1 d: Execução de "Incluir Arco" em EDIT
para grafos do tipo 1 ou 2

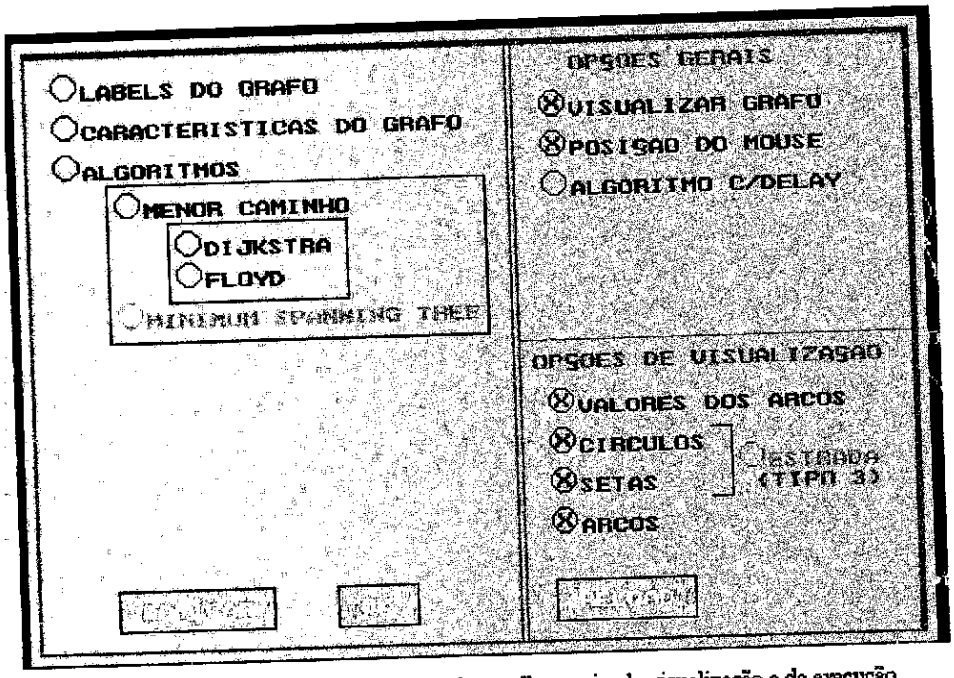


Figura 2: Menu OPTION e as opções de escolha gerais, de visualização e de execução

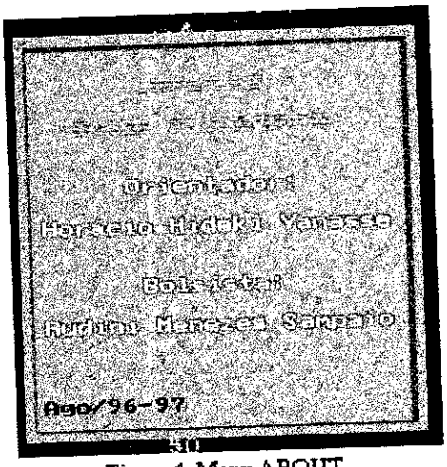


Figura 4: Menu ABOUT

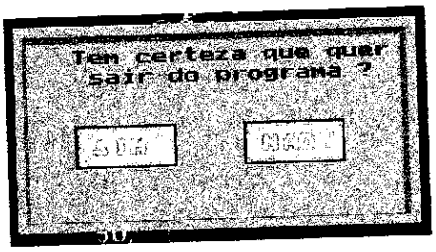


Figura 5: Menu EXIT

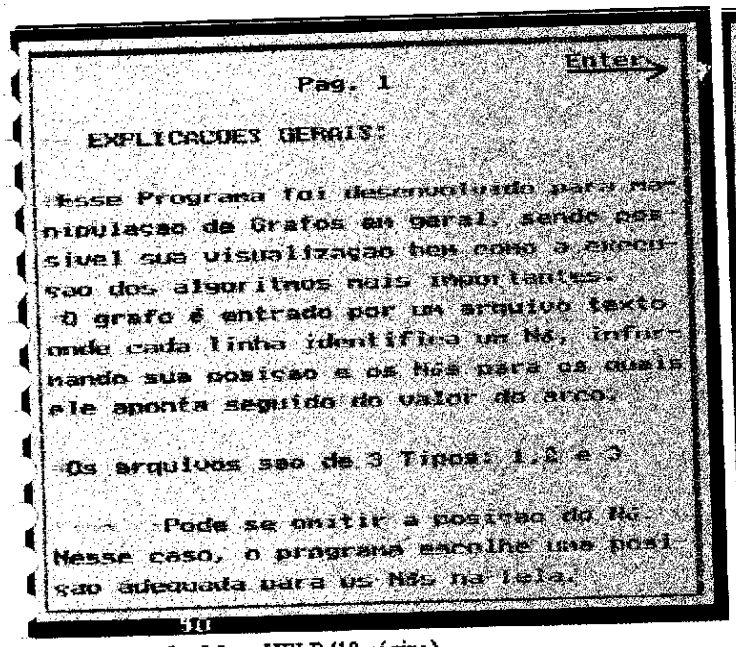


Figura 3a: Menu HELP (1° página)

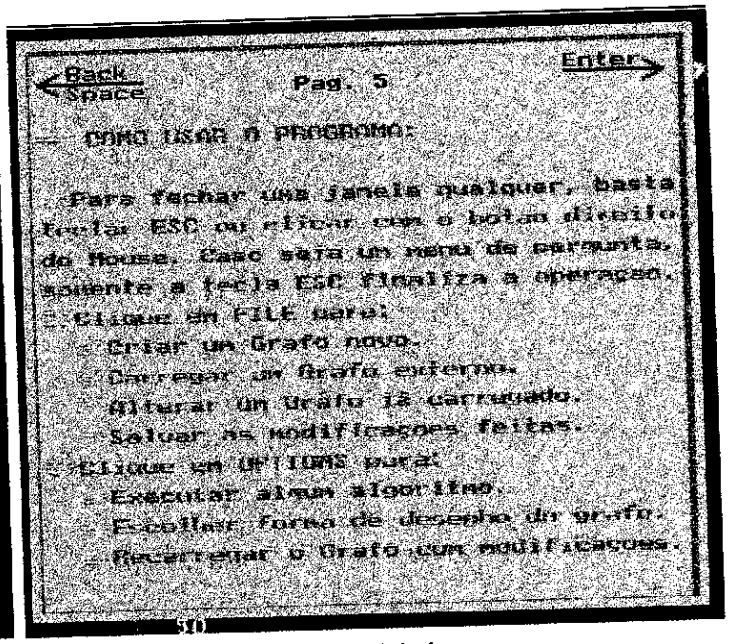
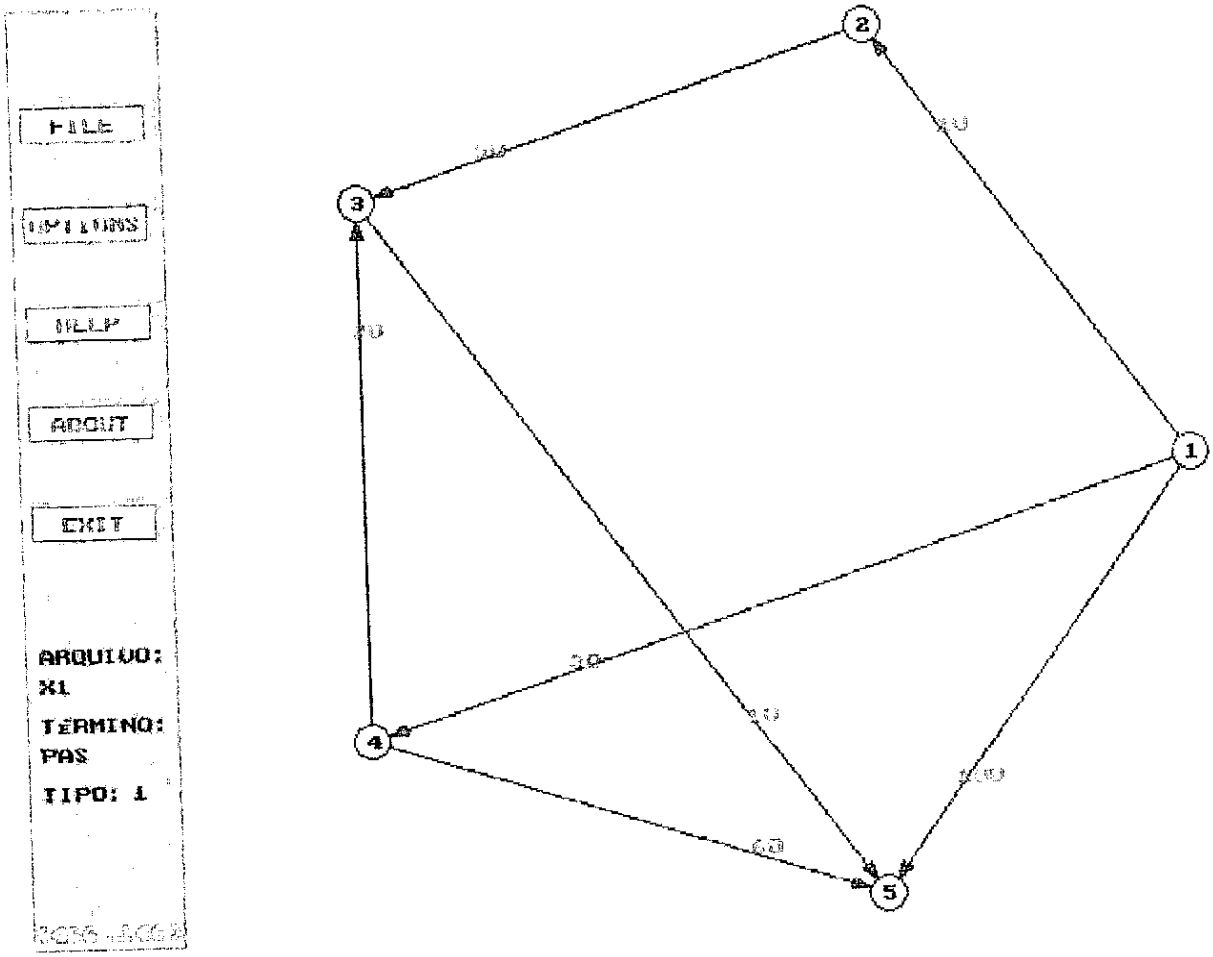


Figura 3b: Menu HELP (5° página)



TIPO1
2 10 4 30 5 100 0 Sao Paulo
3 50 0 Rio de Janeiro
5 10 0 Fortaleza
3 20 5 60 0 Sao Joao Del Rey
0 Miracema do Norte

Figura 6: Arquivo de entrada para exemplificação dos algoritmos de Dijkstra e Floyd

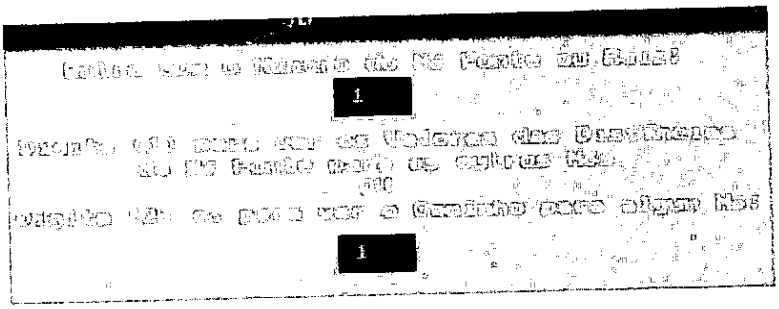


Figura 7a: Mensagem para escolha de visualização do algoritmo de Dijkstra

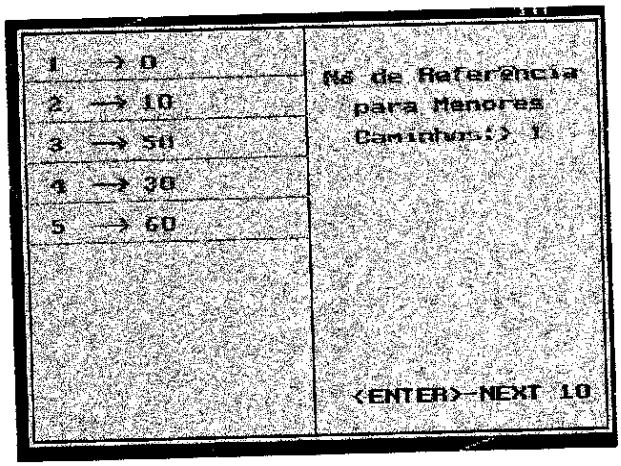


Figura 7b: Valores dos menores caminhos mostrados segundo as opções feitas na figura 7a

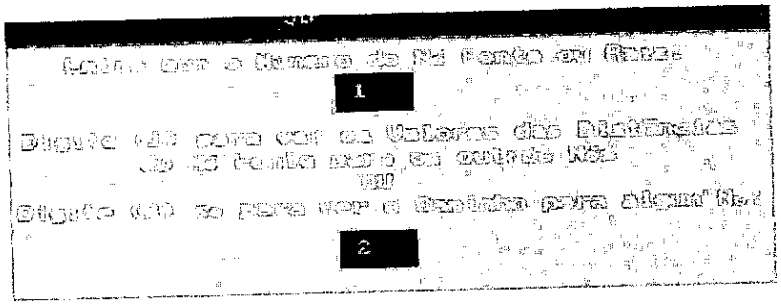


Figura 7c: Mensagem para visualização do algoritmo de Dijkstra

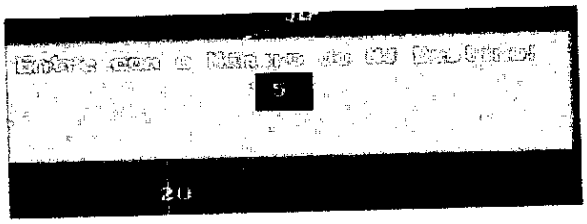


Figura 7d: Mensagem pedindo Nó para ser mostrado o menor caminho em relação as escolhas feitas na figura 7c

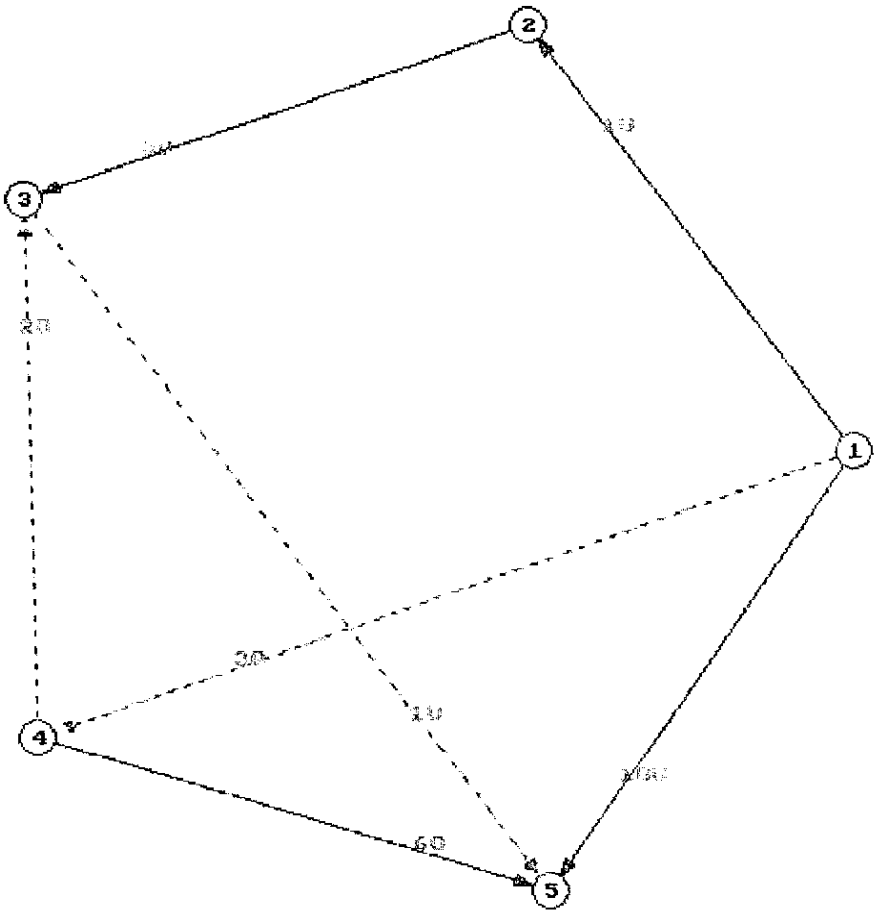


Figura 7e: Menor caminho entre os nós 1 e 5, escolhidos na figura 7d para o algoritmo de Dijkstra

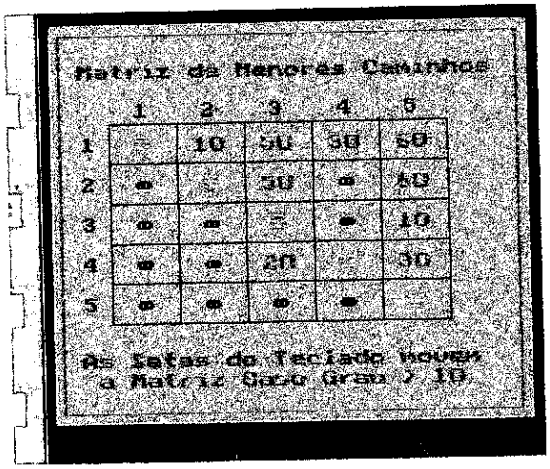


Figura 8: Matriz de menores caminhos, obtida a patir do algoritmo de Floyd.

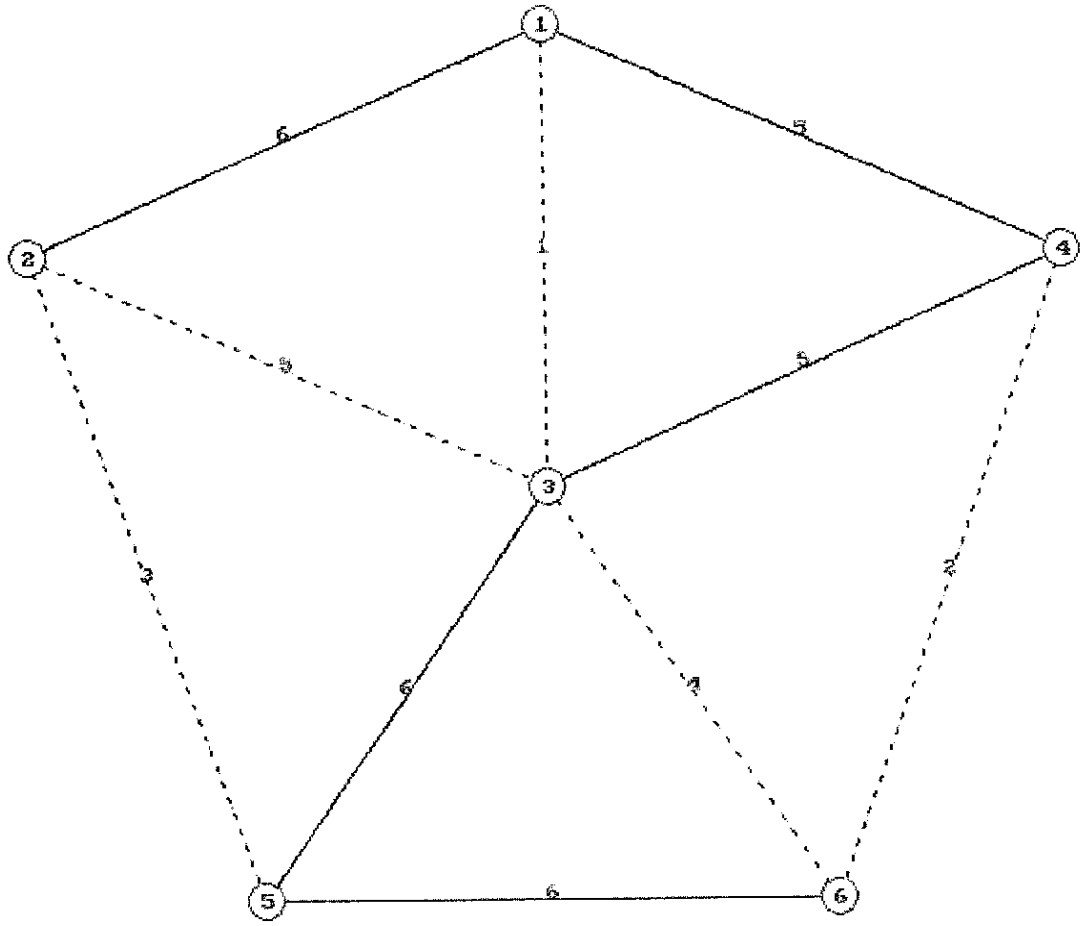


Figura 9: Exemplo de execução do algoritmo de Prim para a árvore de custo mínimo de um grafo direcionado e conexo

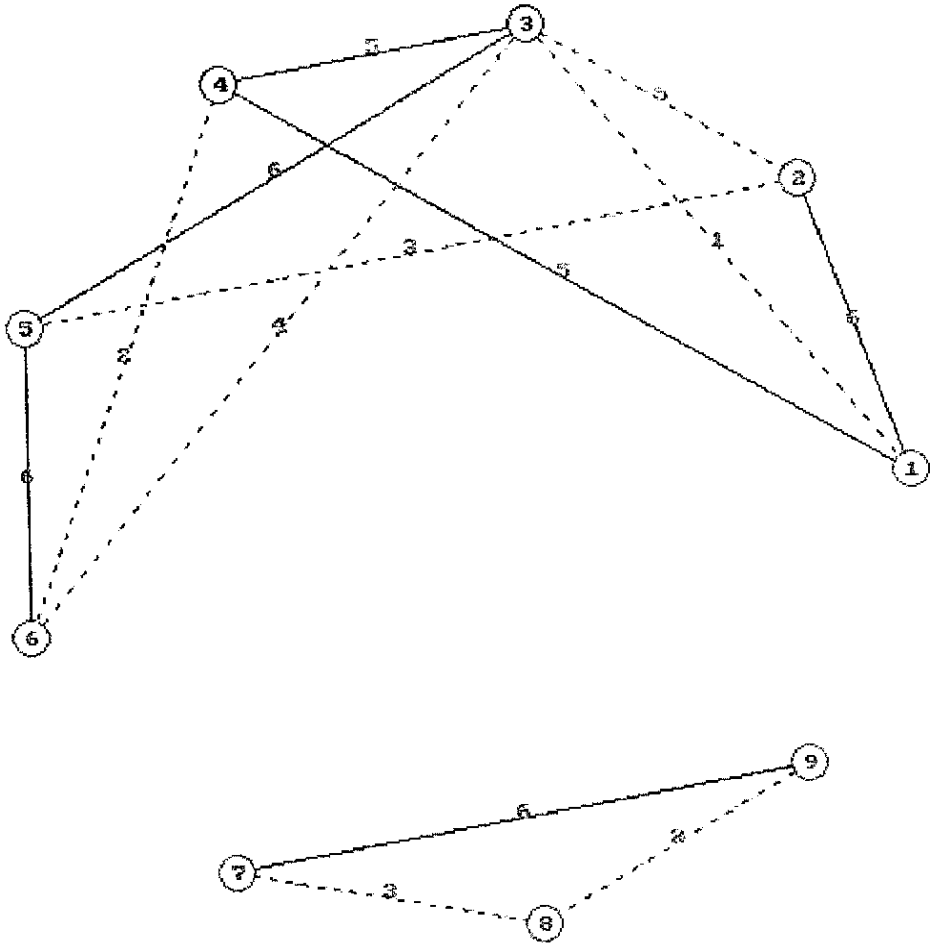


Figura 10: Exemplo de execução do algoritmo de Prim para a árvore de custo mínimo de um grafo direcionado e desconexo

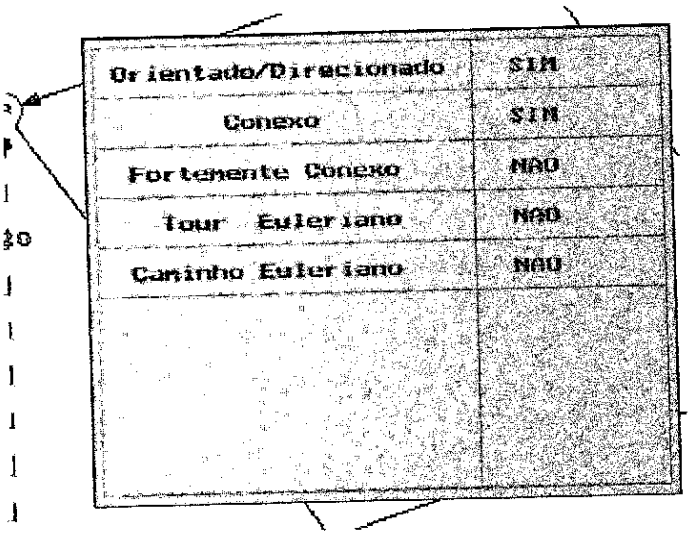


Figura 11a: Características do grafo da figura 5.
Conexo

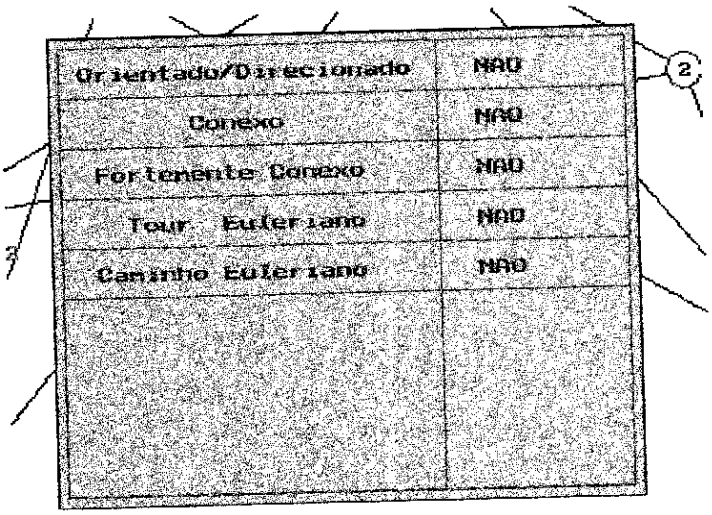


Figura 11c: Características do grafo da figura 7.
Conexo

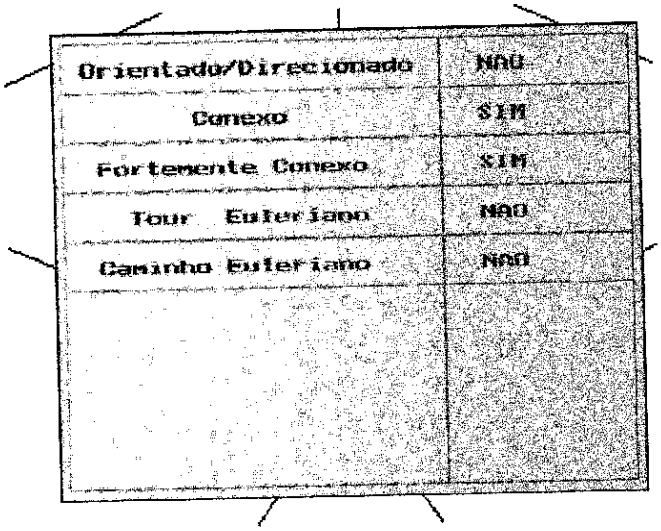


Figura 11b: Características do grafo da figura 6.
Conexo

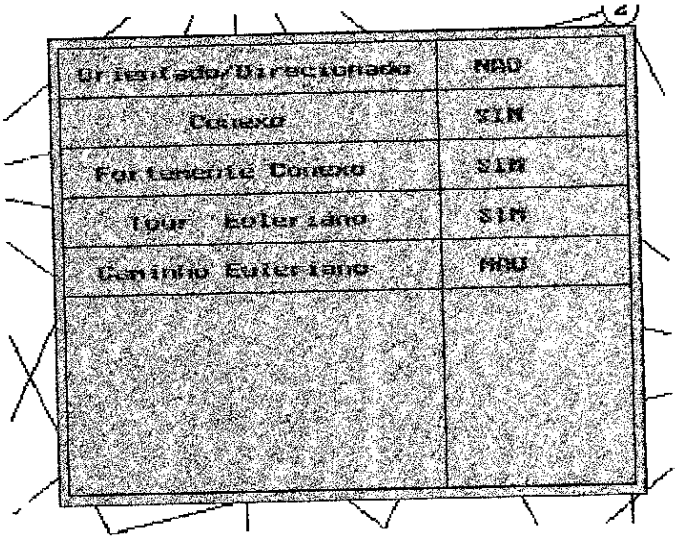
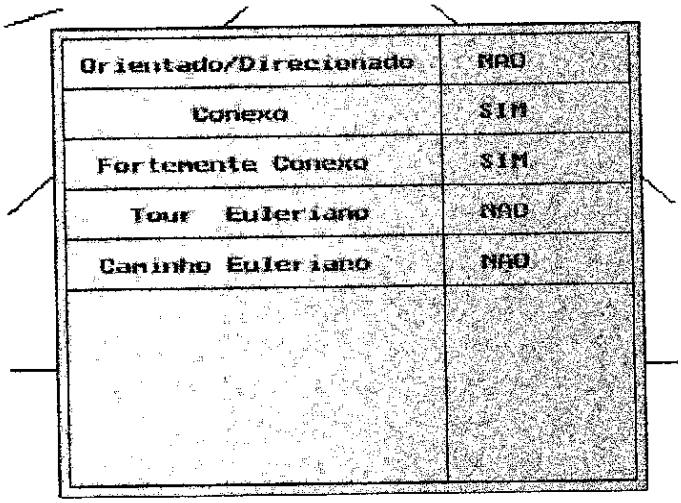
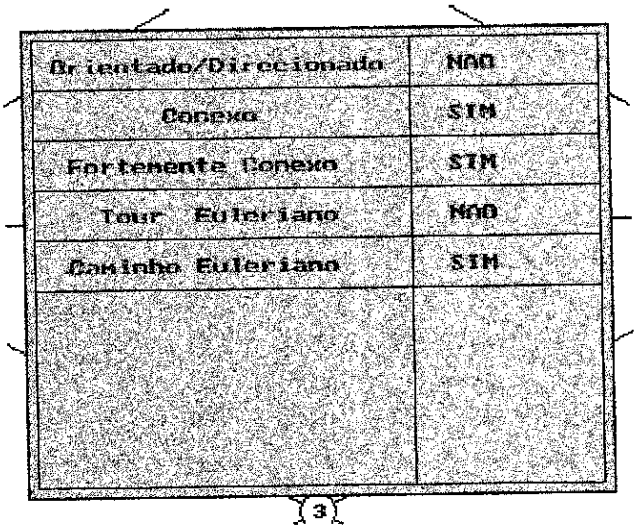


Figura 11d: Características do grafo da figura 12.
Tour Euleriano



Orientado/Direcionado	NAO
Conexo	SIM
Fortemente Conexo	SIM
Tour Euleriano	NAO
Caminho Euleriano	NAO

Figura 11e: Características do grafo da figura 13. Tour e Caminho Euleriano



Orientado/Direcionado	NAO
Conexo	SIM
Fortemente Conexo	SIM
Tour Euleriano	NAO
Caminho Euleriano	SIM

Figura 11f: Características do grafo da figura 14. Caminho Euleriano

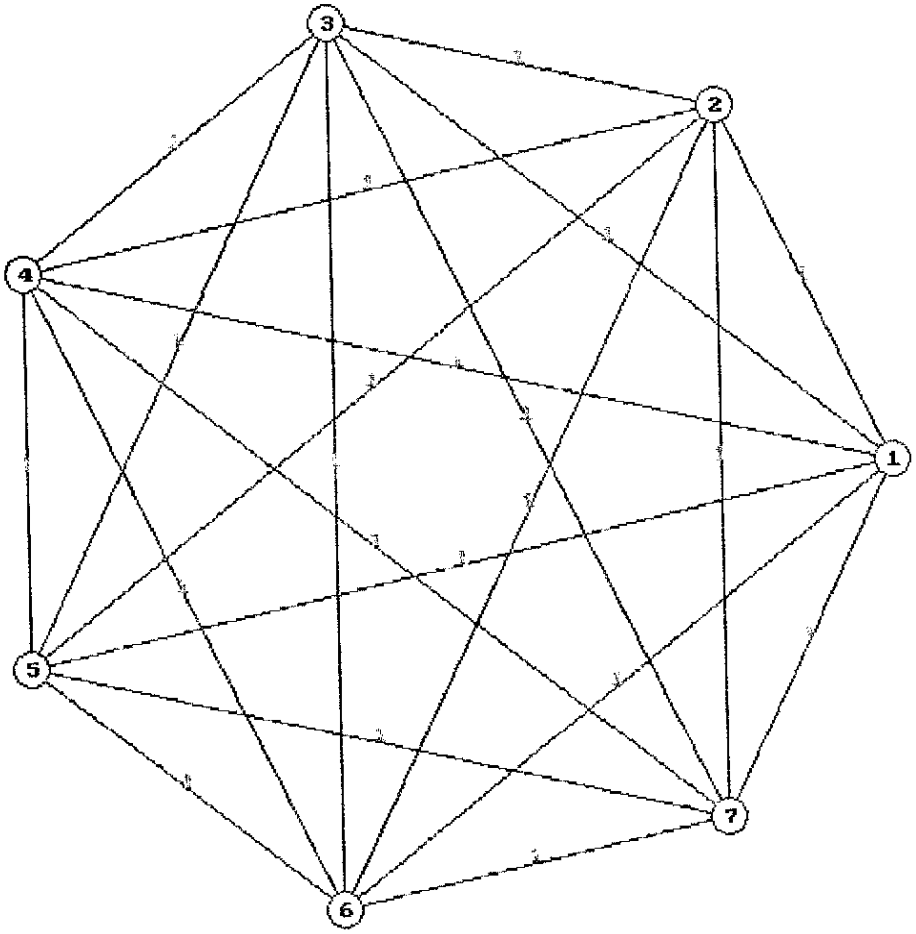


Figura 12: Grafo para exemplificação de algoritmos.

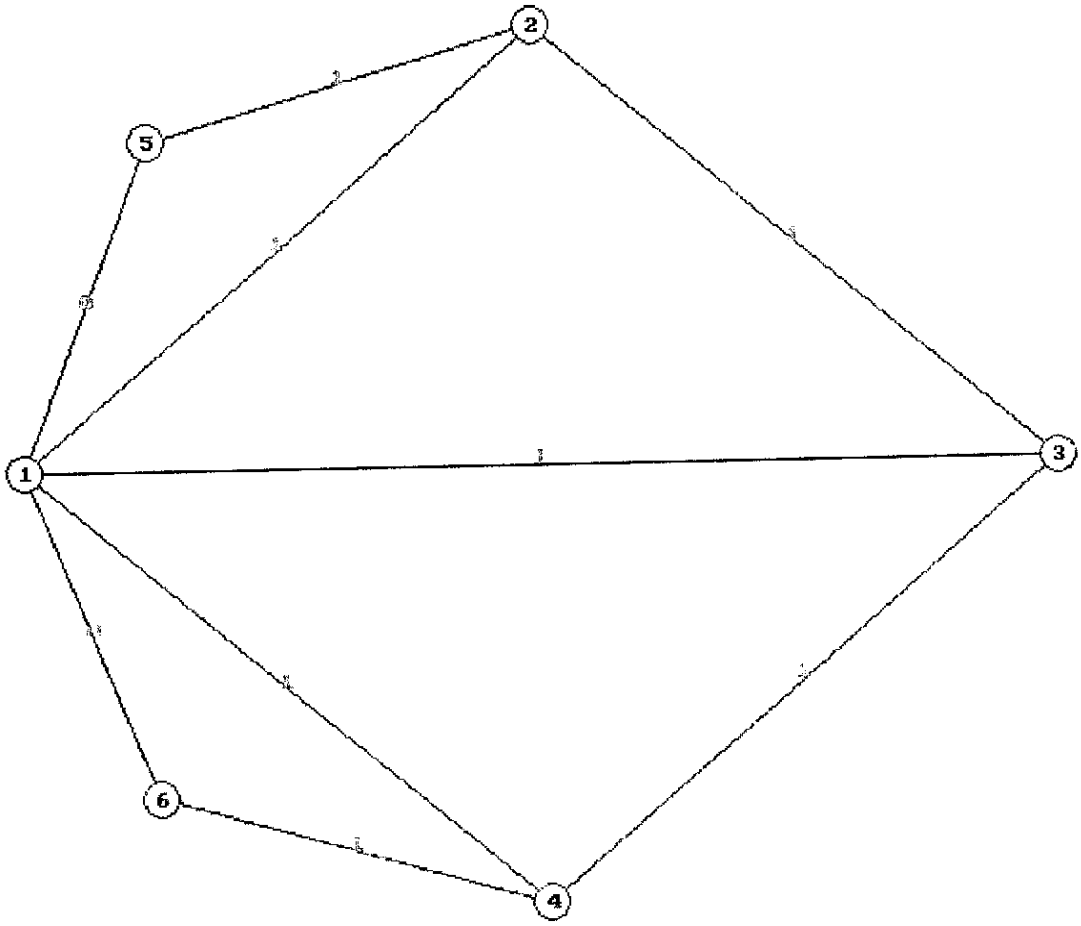


Figura 13: Grafo para exemplificação de algoritmos
(Corresponde ao problema das 7 pontes de Königsberg)

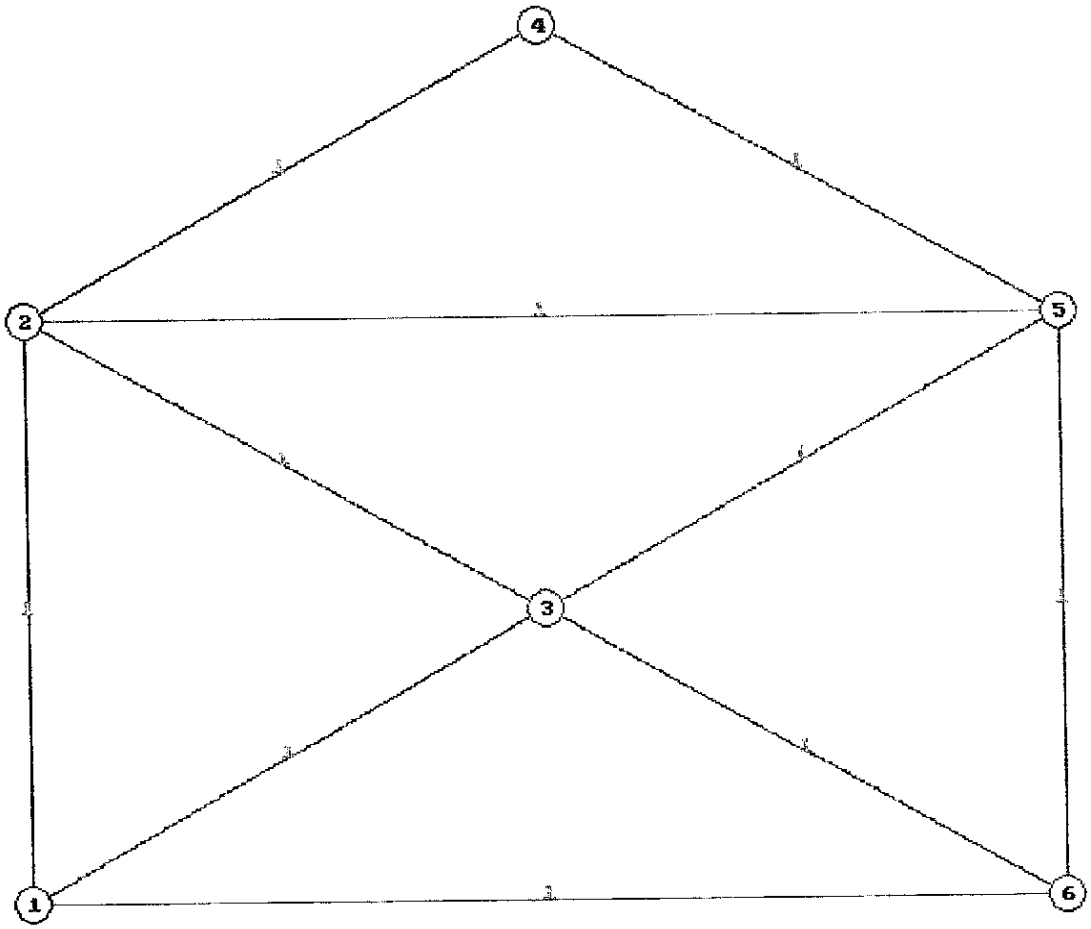


Figura 14: Grafo para exemplificação dealgoritmos (este possui caminho euleriano)

```
PROGRAM GRAFOS;
{$I-}
{$M 65520,0,655360}

USES CRT,GRAPH,DOS,MOUSE,GERAL,FILE_,OPTION,JAN345,DESENHA,IMPLEME
N;

{    }PROCEDURE JANELA(Blok:ShortInt);
     var a,b,c,d:integer;
     begin
     MouseCursor(On);
     Case Blok of
          01:begin a:=90 ;b:=8 ;c:=202;d:=240 end;{ File   }
          02:begin a:=95 ;b:=8 ;c:=550;d:=320 end;{Options}
          03:begin a:=100;b:=8 ;c:=440;d:=300 end;{ Help   }
          04:begin a:=100;b:=98;c:=302;d:=300 end;{ About  }
          05:begin a:=100;b:=98;c:=302;d:=200 end;{ Exit   }
          end;
     SetMouse(700,500) ;
     Grava_Cola(10)    ;
     SetFillStyle(1,7) ;
     Bar(a,b,c,d)      ;
     SetColor(1)       ;
     Rectangle(a+5,b+7,c-7,d-5);
     Rectangle(a+5,b+6,c-6,d-5);
     Rectangle(a+5,b+5,c-5,d-5);
     MWindow  (a+5,b+7,c-7,d-5);
     Case Blok of
          01:FILES   ;
          02:OPTIONS ;
          03:HELP    ;
          04:ABOUT   ;
          05:EXIT_   ;
          end;
     SetColor(0);
     MWindow(0,0,640,480);
     MouseCursor(Off);
     Grava_Cola(01)  ;
     MouseCursor(On) ; SetColor(CT);
     if NewLoad then
        begin NewLoad:=False;
        MouseCursor(Off);
        DSN_ARC_ENCAD   ;
        MouseCursor(On) ;
        Reload:=True    ;
        if (Orientad)and(TipAlg=3) then TipAlg:=0;
        end;
     if (Not JA)and(ArqFixo<>'') then  {JA nao deixa o programa}
```

```
        begin                                   {pentelhar o tempo todo }
        JA:=True;                               {pedindo os Nos do Algoritm
o}
        if BLabel then VerLabels;
        if BCarac then VerCaract;
        if Algor  then
           begin
           if (NFntAnt<>0)and(TipAlg=1) then
               InterLiga(XYapg,NFntAnt,NDstAnt,10); {Dijk}
           if (TipAlg=3)and(Not(Reload)) then       {SpanTr}
               begin
               MouseCursor(Off);
               DSN_ARC_ENCAD;
               MouseCursor(On);
               Reload:=False;
               end;
            SetColor(CT);
            if MenorCam then Menor_Caminho ;
            if (SpanTr)and(not(Orientad)) then
               begin
               Spanning_Tree ;
               Reload:=False;
               end;
            end;
         end;
      end;

  {PROGRAMA PRINCIPAL}
  BEGIN
  {}
  {Declaracoes}
  BMouse    := True;
  BVal      := True;
  BCirc     := True;
  BSeta     := True;
  BArc      := True;
  BVerGrf   := True;
  BEstr     := False;
  BAtraso   := False;
  Algor     := False;
  MenorCam  := False;
  SpanTr    := False;
  Dijk      := False;
  FLd       := False;
  NewLoad   := False;
  Modify    := False;
  BLabel    := False;
  Orientad  := True ;
```

```
ArqFixo  :=  ''  ;
ArqAnt   :=  ''  ;
TipAlg   :=  0   ;
NFntAnt  :=  0   ;
NDstAnt  :=  0   ;
for i:=1 to Tam do XYapg[i]:=0;
arquivo:='arqcop.pas';
{Fim Declaracoes}
CF:=0;     {Cor de Fundo                      (00) }
CT:=14;    {Cor do Traco                      (11) }
CArc:=9;   {Cor do Arco                       (09) }
CCirc:=7;  {Cor dos Circulos dos Nos          (07) }
CCust:=10;{Cor do Custo do Arco               (10) }
CMen:=3;   {Cor do Menu da Apresentacao       (03) }
{
 1) A Indicacao de Cores significam:
          00---PRETO              | 08---CINZA ESCURO
          01---AZUL               | 09---AZUL CLARO
          02---VERDE              | 10---VERDE CLARO
          03---CIANO              | 11---CIANO CLARO
          04---VERMELHO           | 12---VERMELHO CLARO
          05---MAGENTA            | 13---MAGENTA CLARO
          06---MARROM             | 14---AMARELO
          07---CINZA CLARO        | 15---BRANCO
 }
 Randomize;
 GD:=detect;
 InitGraph(GD,GM,'');
 APRESENTACAO; Car:=' ';
 MouseCursor(On);
 SetMouse(320,240);
 vXmouse:=0; vYmouse:=0;
 repeat if BMouse then SetColor(15) else SetColor(CMen);
        Str(vXmouse,aux1);  Str(vYmouse,aux2);
        OutTextXY (4 ,465,'('+aux1);
        OutTextXY (38,465,','+aux2);
        OutTextXY (71,465,')'     );
        if BMouse then
           begin
           SetColor(CMen);
           if (vXmouse<>GetMouseX) then begin OutTextXY(12,465,aux1
 );vXmouse:=GetMouseX end;
           if (vYmouse<>GetMouseY) then begin OutTextXY(46,465,aux2
 );vYmouse:=GetMouseY end;
           end;
        if((button=1)and(GetMouseX>80)) then Zoom;
        if KeyPressed then Car:=readkey;
        if Car=#0 then
```

```
          begin Car:=readkey;
          if Car=#59 then    {F1 - Help}
             begin
             Car:=' '; rebaixo(3);
             OutTextXY(24,159,'HELP');
             Janela(3); Car:=' ';
             end;
          end;
       if button=1 then
          begin
          if MouseIn(8,53 ,70, 70) then
             begin rebaixo(1);OutTextXY(24,59 , 'FILE'   ) ;Janela(
1) end;
          if MouseIn(8,103,70,120) then
             begin rebaixo(2);OutTextXY(12,109,'OPTIONS') ;Janela(
2) end;
          if MouseIn(8,153,70,170) then
             begin rebaixo(3);OutTextXY(24,159, 'HELP'   ) ;Janela(
3) end;
          if MouseIn(8,203,70,220) then
             begin rebaixo(4);OutTextXY(20,209, 'ABOUT'  ) ;Janela(
4) end;
          if MouseIn(8,253,70,270) then
             begin rebaixo(5);OutTextXY(24,259, 'EXIT'   ) ;Janela(
5) end;
          Car:=' ';
          end;
       if button=2 then Car:=#27;
       if Car=#27 then
          begin
          Car:=' '; rebaixo(5);
          OutTextXY(24,259,'EXIT');
          Janela(5);
          end;
       until False;
Finalizar(0);
end.

{Exemplo do Funcionamento do Encadeamento}
{
writeln('');
for i:=1 to N do
    begin
    p:=grf.espvert[i].lista;
    while p<>nil do
          begin
          write(p^.vertice,' ');
          write(p^.arco,' ');
```

```
            p:=p^.prox;
            end;
      writeln;
      end;
writeln('');{}
```

```
UNIT FILE_;
{$I-}
INTERFACE

USES CRT,GRAPH,DOS,MOUSE,GERAL;

VAR Entrou:Boolean;
    arq   :text   ;
    erro  :integer;

PROCEDURE FILES;

IMPLEMENTATION

FUNCTION SUBSTITUIR:Boolean;
   var arq1:string;
       e:char;
   begin
   Substituir:=False   ;
   SetColor(1); SS:='' ;
   SetFillStyle(1 , 7) ;
   Bar(210,25,440,165) ;
   Rectangle(215,30,435,160);
   OutTextXY(230,40,'Entre o Nome do Arquivo:');
   SetFillStyle(1 , 0) ;
   Bar(265,55,380,80);
   LerStr(arq1,275,63,12);
   if arq1='' then begin SetColor(0); MouseCursor(On); exit end;
   if (arq1<>'') then SS:=FSearch(arq1,GetEnv('PATH'));
   if SS='' then Substituir:=True
   else begin
        OutTextXY(235,90,'Arquivo j'+aa+' existente');
        OutTextXY(235,105,'Deseja substitui-lo?');
        OutTextXY(235,120,' ( Sim(S) / Nao(N) )');
        SetColor(7);
        repeat Bar(300,135,330,155);
               e:=upcase( readkey );
               if ord(e)>32 then OutTextXY(312,142,e);
               if (e<>'S')and(e<>'N')and(e<>#27) then begin beep;
 delay(300) end;
               until (e='S')or(e='N')or(e=#27);
        if e='S' then Substituir:=True;
        end;
    if (SS='')or(e='S') then ArqFixo:=arq1;
    SetColor(0);
    end;
 PROCEDURE SAIDA ;
   begin
```

```
  Close(arq);
  SetColor(0);
  Grava_Cola(01);
  Grava_Cola(10);
  end;
FUNCTION INCNO(Strin:string) : Boolean  ;
  var XNoS,YNoS,aux:string;    {Strin diz se quem chamou a rotina}
      XNoI,YNoI:integer;       {     foi EDIT ou se foi NEW      }
      tst,err  :integer;
  begin
  IncNo:=False;
  append(arq);
  if (Strin='edit') then Barra(110, 50,2,' Incluir N'+oo,120,0);
  if (Tipo in[2,3]) then
      begin
      Barra(310,60,7,'',210,120);
      SetColor(1);
      Rectangle(315,65,515,175);
      OutTextXY(320,74,' Entre com X e Y do No  ');
      OutTextXY(320,89,'a ser incluido no grafo:');
      OutTextXY(380,117,'X:'); OutTextXY(465,117,'*');
      OutTextXY(380,147,'Y:');
      Barra(400,108,0,'',60,20);
      Barra(400,140,0,'',60,20);
      LerInt(XNoI,XNoS,410,116,5);
      if XNoI=0 then begin Saida; exit end;
      SetColor(1);  OutTextXY(465,147,'*');
      SetColor(7);  OutTextXY(465,117,'*');
      LerInt(YNoI,YnoS,410,148,5);
      if YNoI=0 then begin Saida; exit end;
      writeln(arq,XNoS+' '+YNoS+' 0');
      end
   else writeln(arq,'0');
   if (Strin='edit') then
      begin
      Str(N+1,aux);
      Barra(310,185,7,'',100,50);
      SetColor(1);
      Rectangle(315,190,405,230);
      OutTextXY(340,195,'  OK');
      OutTextXY(340,215,aux+' N'+oo+'s');
      delay(500);
      end;
   N:=N+1;
   IncNo:=True;
   Saida;
   end;
  {}
```

```
FUNCTION EXCNO : Boolean  ;
  var No,Ji,Ki,i,l:integer;
      Js:string[4];
      Ks,wrd:string ;
      lixo:array[1..Tam]of string;
  begin
  ExcNo:=False;
  Reset(arq);
  Barra(110, 90,2,' Excluir N'+oo,120,0);
  Barra(310,60,7,'',220,90);
  SetColor(1); Str(N+1,wrd );
  Rectangle(315,65,525,145 );
  OutTextXY(320,74,'Abaixo, entre com o valor');
  OutTextXY(320,89,'  do N'+oo+' a ser retirado:  ');
  OutTextXY(375,117,'N'+oo+':  ');
  Barra(400,108,0,'',60,20);
  SetColor(0);
  Str(N,Js);
  repeat LerInt(No,Ks,410,116,length(Js));
         if No=0 then begin Saida; exit end;
         if No>N then
            begin
            OutTextXY(480,117,'N'+oo+'<'+wrd);
            Barra(400,108,0,'',60,20);
            end;
         until No<=N;
   readln(arq,wrd);
   lixo[1]:=wrd; l:=1;
   for i:=1 to N do
       begin
       if i<>No then
          begin
          l:=l+1;
          if Tipo<>1 then
             begin
             read(arq,Ji,Ki); Str(Ji,Js);
             Str(Ki,Ks); wrd:=Js+' '+Ks+' ';
             end
          else wrd:=''; Ji:=1;
          while Ji<>0 do
                begin
                read(arq,Ji); if Ji>No then Ki:=1 else Ki:=0;
                Str(Ji-Ki,Js);
                if     Ji<>No      then wrd:=wrd+Js ;
                if not(Ji in[0,No])then wrd:=wrd+' ';
                if (Tipo<>3)and(Ji<>0) then
                   begin
                   read(arq,Ki); Str(Ki,Ks);
```

```
                    if Ji<>No then wrd:=wrd+Ks+' ';
                    end;
                end;
          readln(arq,Ks);
          lixo[1]:=wrd+Ks;
          end
       else readln(arq);
       end;
  close(arq); rewrite(arq);
  for i:=1 to N do writeln(arq,lixo[i]);
  N:=N-1;
  ExcNo:=True;
  Saida;
  end;
{}
FUNCTION INC_EXC_ARC(opcao:integer) : Boolean;
  var NoFi,NoDi,ArcI,Ji,Ki,i:integer;
      Js:string[4];
      NoFs,NoDs,ArcS,Ks,wrd:string ;
      lixo:array[1..Tam]of string;
  begin
  Inc_Exc_Arc:=False;
  Reset(arq);
  Barra(310,60,7,'',270,150);
  SetColor(1); Str(N+1,wrd);
  Rectangle(315,65,575,205);
  OutTextXY(320,74,' Abaixo, entre com o valor do');
  if (opcao=01)or(Tipo=3)then OutTextXY(345,89,'No Fonte e do No D
estino:')
  else begin
       OutTextXY(320,89,'No Fonte, No Destino e do Arco:');
       OutTextXY(368,177,'Arco:');
       Barra(420,172,0,' ',60,20);
       SetColor(1);
       end;
   OutTextXY(365,117,'Fonte:  '); OutTextXY(485,117,'*');
   OutTextXY(355,147,'Destino:');
   Barra(420,108,0,'',60,20);
   Barra(420,140,0,'',60,20);
   SetColor(0);
   str(N,Js);
   repeat LerInt(NoFi,NoFs,430,116,Length(Js));
          if NoFi=0 then begin Saida; exit end;
          if NoFi>N then
             begin
             OutTextXY(500,117,'N'+oo+'<'+wrd);
             Barra(420,108,0,'',60,20);
             end;
```

```
       until NoFi<=N;
SetColor(7);  OutTextXY(485,117,'*');
OutTextXY(500,117,'N'+oo+'<'+wrd);
SetColor(1);  OutTextXY(485,147,'*'); SetColor(0);
Str(N,Js);
repeat LerInt(NoDi,NoDs,430,148,Length(Js));
       if NoDi=0 then begin Saida; exit end;
       if NoDi>N then
          begin
          OutTextXY(500,147,'N'+oo+'<'+wrd);
          Barra(420,140,0,'',60,20);
          end;
       until NoDi<=N;
SetColor(7);  OutTextXY(485,147,'*');
OutTextXY(500,147,'N'+oo+'<'+wrd);
SetColor(1);
if opcao=10 then
   if Tipo<>3 then
      begin
      OutTextXY(485,177,'*');
      LerInt(ArcI,ArcS,430,180,5);
      if Arcs=' ' then begin Saida; exit end;
      end
   else ArcS:='';
for i:=1 to NoFi do
    begin
    readln(arq,wrd);
    lixo[i]:=wrd;
    end;
if Tipo<>1 then
   begin
   read(arq,Ji,Ki); Str(Ji,Js);
   Str(Ki,Ks); wrd:=Js+' '+Ks+' ';
   end
else wrd:='';       Ji:=1;
if (opcao=10) then wrd:=wrd+NoDs+' '+ArcS+' ';
while Ji<>0 do
       begin
       read(arq,Ji); Str(Ji,Js);
       if Ji<>NoDi then wrd:=wrd+Js ;
       if Ji<> 0   then wrd:=wrd+' ';
       if (Tipo<>3)and(Ji<>0) then
          begin
          read(arq,Ki); Str(Ki,Ks);
          if Ji<>NoDi then wrd:=wrd+Ks+' ';
          end;
       end;
readln(arq,Ks);
```

```
  lixo[NoFi+1]:=wrd+Ks;
  for i:=NoFi+2 to N+1 do
      begin
      readln(arq,wrd);
      lixo[i]:=wrd;
      end;
  close(arq);
  rewrite(arq);
  for i:=1 to N+1 do writeln(arq,lixo[i]);
  Inc_Exc_Arc:=True;
  Saida;
  end;
{}
FUNCTION INCARC : Boolean ;
  begin
  Barra(110,130,2,'Incluir Arco',120,0);
  INCARC:=INC_EXC_ARC(10);
  end;
{}
FUNCTION EXCARC : Boolean ;
  begin
  Barra(110,170,2,'Excluir Arco',120,0);
  EXCARC:=INC_EXC_ARC(01);
  end;
{}
FUNCTION MUDLAB :Boolean;
  var wrd,Ks:string;
      Js:string[4];
      No,Ji,Ki,i,l:integer;
      lixo:Array[1..Tam]of string;
  begin
  MudLab:=False;
  Barra(110,210,2,'Mudar Labels',120,0);
  Barra(310,60,7,'',240,90);
  SetColor(1); Str(N+1,wrd );
  Rectangle(315,65,545,145 );
  OutTextXY(320,74,'Abaixo, entre com o valor do');
  OutTextXY(320,89,'  N'+oo+' cujo Label sera mudado:');
  OutTextXY(375,117,'N'+oo+':  ');
  Barra(400,108,0,'',60,20);
  SetColor(0);
  Str(N,Js);
  Reset(arq);
  repeat LerInt(No,Ks,410,116,length(Js));
         if No=0 then begin Saida; exit end;
         if No>N then
            begin
            OutTextXY(480,117,'N'+oo+'<'+wrd);
```

```
            Barra(400,108,0,'',60,20);
            end;
        until No<=N;
 Barra(305,60,7,'',250,90);
 Rectangle(310,65,550,145 );
 SetColor(1); wrd:=copy(Labels[No],2,18);
 OutTextXY(320,74 ,'Abaixo, entre com o Label do');
 OutTextXY(320,89 ,' N'+oo+' escolhido anteriormente:');
 OutTextXY(330,130,'Default:'+wrd);
 OutTextXY(330,112,'Label:');
 Barra(380,105,0,'',160,20);
 SetColor(0);
 LerStr(wrd,390,113,18);
 if wrd='' then begin Saida; exit end;
 Labels[No]:=wrd;
 readln(arq,wrd);
 lixo[1]:=wrd;
 for i:=1 to N do
     if ( i<>No ) then
          begin
          readln(arq,wrd);
          lixo[i+1]:=wrd;
          end
     else begin
          Ji:=1; wrd:='';
          while Ji<>0 do
                begin
                read(arq,Ji); Str(Ji,Js);
                wrd:=wrd+Js+' ';
                end;
          wrd:=wrd+labels[No];
          lixo[i+1]:=wrd;
          readln(arq);
          end;
  close(arq); rewrite(arq);
  for i:=1 to N+1 do writeln(arq,lixo[i]);
  Labels[No]:=' '+Labels[No];
  MudLab:=True;
  Saida;
  end;
{}
PROCEDURE NEW ;
   var XXs,YYs,TipoS:string;
       XXi,YYi:integer;
       Opt:Char;
   begin
   SALV_ALTERACOES;
   if Substituir then
```

```
      begin
      NewLoad:=True;
      SetColor(1);
      SetFillStyle(1 , 7) ;
      Bar(210,175,440,260) ;
      Rectangle(215,180,435,255);
      OutTextXY(230,190,'Entre o Tipo do Arquivo:');
      OutTextXY(230,205,'Valores Possiveis: 1 2 3');
      SetFillStyle(1 , 0) ;
      repeat Bar(310,220,330,240);
             LerInt(Tipo,TipoS,317,227,1);
             if not(Tipo in [1,2,3]) then Beep;
             until Tipo in[1,2,3];
      Assign(arq,ArqFixo);
      ReWrite(arq);
      Writeln(arq,'TIPO'+TipoS);
      Saida;
      repeat until IncNo('new');
      copia(ArqFixo,arquivo);
      end;
   MouseCursor(On);
   end;
{}
PROCEDURE LOAD;
   var arq1:string; e:char;
   begin
   SALV_ALTERACOES;
   SetColor(1); SS:='' ;
   SetFillStyle(1,7)   ;
   Bar(210,75,440,170) ;
   Rectangle(215,80,435,165);
   OutTextXY(230,89,'Entre o Nome do Arquivo: ');
   SetFillStyle(1,0)   ;
   Bar(265,105,380,130);
   LerStr(arq1,275,113,12);
   if arq1='' then begin SetColor(0); MouseCursor(On); exit end;
   if (arq1<>ArqFixo) then SS:=FSearch(arq1,GetEnv('PATH'));
   if SS<>'' then begin ArqFixo:=arq1; NewLoad:=True end
   else begin
        SetColor(4);
        if (arq1<>ArqFixo) then aux1:='Arquivo nao encontrado'
        else aux1:=' Arquivo j'+aa+' carregado';
        OutTextXY(235,140,aux1);
        Mwindow(215,80,435,165); Beep;
        SetMouse(230,100); MouseCursor(On);
        repeat if keypressed then car:=readkey;
               until (car=#27)or(button=2);
        end;
```

```
   SetColor(0)     ;
   MouseCursor(On);
   end;
{}
PROCEDURE SAVE;
   begin
   if ArqFixo<>'' then copia(arquivo,ArqFixo);
   MouseCursor(On);
   Modify:=False;
   end;
{}
PROCEDURE SAVE_AS;
  begin
  if (ArqFixo<>'')and(Substituir) then
     begin
     SetColor(CMen); Barra(10,337,CMen,'',65,20);
     SetColor(CMen); Barra(10,370,CMen,'',65,20);
     Copia(arquivo,ArqFixo);
     ESC_NOME_ARQ;
     end;
  MouseCursor(On)
  end;
PROCEDURE EDIT;
  begin
  if ArqFixo='' then begin MouseCursor(On); exit end;
  Assign(arq,arquivo);
  Entrou:=True;
  Car:=' ';
  SetColor(0);
  repeat if Entrou then
            begin
            SetColor(1);
            SetFillStyle(1,7);
            Bar(90,8,262,270);
            Rectangle(95,13,257,265);
            Mwindow(95,13,257,265);
            Barra(110,50 ,9,' Incluir N'+oo,120,0);
            Barra(110,90 ,9,' Excluir N'+oo,120,0);
            Barra(110,130,9,'Incluir Arco' ,120,0);
            Barra(110,170,9,'Excluir Arco' ,120,0);
            Barra(110,210,9,'Mudar Labels' ,120,0);
            MouseCursor(On);
            end;
         Entrou:=False; Car:=' ';
         if button=1 then
            begin
            MouseCursor(Off);
            if MouseIn(110, 50,350, 70)and(IncNo('edit'))then Modi
```

```
fy:=True;
            if MouseIn(110, 90,350,110)and(    ExcNo    )then Modi
fy:=True;
            if MouseIn(110,130,350,150)and(    IncArc   )then Modi
fy:=True;
            if MouseIn(110,170,350,190)and(    ExcArc   )then Modi
fy:=True;
            if MouseIn(110,210,350,230)and(    MudLab   )then Modi
fy:=True;
            Repeat until button<>1  ;
            Entrou:=True; Delay(100);
            end;
          if keypressed then Car:=readkey;
          until (Car=#27)or(button=2);
  end;
{}
PROCEDURE FILES;
  begin                                        {a[1]}
  Entrou:=True;{Diz se entrou em New,Load,Save ou Show, ou nao}
  Car:=' ';{Diz qual tecla foi acionada caso alguma tenha sido}
  SetColor(0);
  repeat if Entrou then
            begin
            MouseCursor(Off);
            Barra(115,30 ,9,'NEW ',0,0);
            Barra(115,70 ,9,'LOAD',0,0);
            Barra(115,110,9,'SAVE',0,0);
            Barra(105,150,9,'SAVE AS',75,0);
            Barra(115,190,9,'EDIT',0,0);
            MouseCursor(On);
            end;
          if keypressed then Car:=readkey;
          Entrou:=False;
          if button=1 then
            begin
            MouseCursor(Off);{
---->                                        }
            if MouseIn(115, 30,160, 50) then begin Barra(115, 30,2
,'NEW '   , 0,0) ; NEW     ; exit end  ;
            if MouseIn(115, 70,160, 90) then begin Barra(115, 70,2
,'LOAD'   , 0,0) ; LOAD    ; exit end  ;
            if MouseIn(115,110,160,130) then begin Barra(115,110,2
,'SAVE'   , 0,0) ; SAVE    ; exit end  ;
            if MouseIn(105,150,180,170) then begin Barra(105,150,2
,'SAVE AS',75,0) ; SAVE_AS ; exit end  ;
            if MouseIn(115,190,160,210) then begin Barra(115,190,2
,'EDIT'   , 0,0) ; EDIT    ; exit end  ;
            MouseCursor(On );{
```

```
---->                                                    }
            repeat until button=0;
            Entrou:=True; delay(100);
            end;
          until (Car=#27)or(button=2);
  end;
END.
```

```
UNIT OPTION;
{$I-}
INTERFACE

USES CRT,GRAPH,MOUSE,GERAL;

PROCEDURE OPTINIT(x,y,V:integer;wrd:string);
PROCEDURE OPTCLIC(x,y:integer; var B:Boolean; V:shortint);
PROCEDURE OPTAPAG(x,y:INTEGER; var B:Boolean);
PROCEDURE INIT_OPTIONS;
PROCEDURE OPTIONS;

IMPLEMENTATION

{    }PROCEDURE OPTINIT(x,y,V:integer;wrd:string);
     var B:Boolean;
     begin{Inicia a Tela Options}
     Circle(x,y,6); B:=False;
     OutTextXY(x+9,y,wrd);
     case V of
          1 : if Algor    then B:=True;
          2 : if MenorCam then B:=True;
          3 : if SpanTr   then B:=True;
          4 : if Dijk     then B:=True;
          5 : if FLd      then B:=True;
          6 : if BMouse   then B:=True;
          7 : if BVal     then B:=True;
          8 : if BCirc    then B:=True;
          9 : if BSeta    then B:=True;
          10: if BArc     then B:=True;
          11: if BEstr    then B:=True;
          12: if BVerGrf  then B:=True;
          13: if BAtraso  then B:=True;
          14: if BLabel   then B:=True;
          15: if BCarac   then B:=True;
          end;
     if B then OutTextXY(x-3,y-3,'X');
     end;

{    }PROCEDURE OPTCLIC(x,y:integer; var B:Boolean; V:shortint);
     var T,F:boolean;
         i:integer;
     begin
     T:=True; F:=False;
     if not(B) then
        begin
        Case V of
0 :          ;                                          {QUALQUER}
```

```
1 :begin B:=T;    BLabel  :=F; BCarac:= F ;
         OPTAPAG(120,30,T)  ; OPTAPAG(120,50 ,T) end;{ALGOR}
2 :begin B:=T;    SpanTr  :=F; OPTAPAG(150,146,T) end;{MENOR_CAM}
3 :if not(Orientad) then
   begin B:=T;  MenorCam :=F; OPTAPAG(150,90 ,T) end;{SPANTR}
4 :begin B:=T;    FLd     :=F; OPTAPAG(180,126,T) end;{DIJK}
5 :begin B:=T;    Dijk    :=F; OPTAPAG(180,110,T) end;{FLOYD}
                                                      {BMOUSE}
6 :       B:=T;                                       {BVAL}
7 :       B:=T;
8 :begin B:=T;    BEstr   :=F; OPTAPAG(460,225,T) end;{BCIRC}
9 :begin B:=T;    BEstr   :=F; OPTAPAG(460,225,T) end;{BSETA}
                                                      {BARC}
10:       B:=T;
11:if Tipo=3 then begin B:=T; BCirc:= F  ; BSeta:=F;
         OPTAPAG(365,215,T)  ; OPTAPAG(365,235,T) end;{BESTR}
                                                      {BVERGRF}
12:       B:=T;                                       {BATRASO}
13:       B:=T;
14:begin B:=T;    BCarac  :=F; Algor:= F  ;
         OPTAPAG(120,50,T)  ; OPTAPAG(120,70 ,T) end;{BLABEL}
15:begin B:=T;    BLabel  :=F; Algor:= F  ;
         OPTAPAG(120,30,T)  ; OPTAPAG(120,70 ,T) end;{BCARAC}
end;
        if B then OutTextXY(x-3,y-3,'X');
        end
     else begin
          SetColor(7);
          OutTextXY(x-3,y-3,'X');
          SetColor(0);
          B:=False;
          end;
     end;
{   }PROCEDURE OPTAPAG(x,y:INTEGER; var B:Boolean);
     var B1:Boolean;
     begin
     B1:=B;         {qquer}
     OPTCLIC(x,y,B,0);
     B:=B1;
     end;
{   }PROCEDURE INIT_OPTIONS;
     begin

{02:begin a:=95 ;b:=8 ;c:=465;d:=320 end;{Options}
     Line(342,13,342,315);                  { 245 e -165 }
     Line(345,13,345,315);                  {  75 e -15  }

     SetColor(1);
     Line(345,164,545,164);
     OutTextXY(375,22,'OP□OES GERAIS');
     OutTextXY(352,172,'OP□OES DE VISUALIZA□AO');
```

```
     OutTextXY(455,235,' (TIPO 3)');
     SetColor(0);
     Rectangle(140,80 ,330,160);
     Rectangle(170,100,260,140);
     Line(450,210,450,245);
     Line(440,210,450,210);
     Line(440,245,450,245);
     OPTINIT(120,30 ,14,'LABELS DO GRAFO');
     OPTINIT(120,50 ,15,'CARACTERISTICAS DO GRAFO');
     OPTINIT(120,70 ,1 ,'ALGORITMOS');
     OPTINIT(150,90 ,2 ,'MENOR CAMINHO');
     OPTINIT(180,110 ,4 ,'DIJKSTRA');
     OPTINIT(180,126 ,5 ,'FLOYD');
     if (Orientad) then SetColor(8);
     OPTINIT(150,146,3 ,'MINIMUM SPANNING TREE');
     if (Orientad) then SetColor(0);
     OPTINIT(365,45 ,12,'VISUALIZAR GRAFO');
     OPTINIT(365,65 ,6 ,'POSI□AO DO MOUSE');
     OPTINIT(365,85 ,13,'ALGORITMO C/DELAY');
     OPTINIT(365,195,7 ,'VALORES DOS ARCOS');
     OPTINIT(365,215,8 ,'CIRCULOS');
     OPTINIT(365,235,9 ,'SETAS   ');
     OPTINIT(365,255,10,'ARCOS   ');
     if (Tipo<>3) then  SetColor(8);
     OPTINIT(460,225,11,'ESTRADA ');
     if (Tipo<>3) then  SetColor(0);
     Barra  (250,285,9 ,'OK',40,0 );
     Barra  (140,285,9 ,'APLICAR',78,0);
     Barra  (360,285,9 ,'RELOAD ',70,0);
     end;
{    }PROCEDURE OPTIONS;
     var Entrou:Boolean;
     BEGIN
     Entrou:=True; Car:=' ';
     MouseCursor(Off);
     INIT_OPTIONS;
     MouseCursor(On);
     repeat if keypressed then Car:=readkey;
            Entrou:=False;
            if button=1 then
               begin
               MouseCursor(Off);
               if MouseIn(114,24 ,126,36 )then OPTCLIC(120,30 ,BLA
BEL  ,14);
               if MouseIn(114,44 ,126,56 )then OPTCLIC(120,50 ,BCA
RAC  ,15);
               if MouseIn(114,64 ,126,76 )then OPTCLIC(120,70 ,ALG
OR   ,1 );
```

```
                if MouseIn(144,84 ,156,96 )then OPTCLIC(150,90 ,Men
orCam,2 );
                if MouseIn(174,104,186,116)then OPTCLIC(180,110,Dij
k     ,4 );
                if MouseIn(174,120,186,132)then OPTCLIC(180,126,FLd
      ,5 );
                if MouseIn(144,140,156,152)then OPTCLIC(150,146,Spa
nTr   ,3 );
                if MouseIn(359,39 ,371,51 )then OPTCLIC(365,45 ,BVe
rGrf ,12);
                if MouseIn(359,59 ,371,71 )then OPTCLIC(365,65 ,BMo
use   ,6 );
                if MouseIn(359,79 ,371,91 )then OPTCLIC(365,85 ,BAt
raso ,13);
                if MouseIn(359,189,371,201)then OPTCLIC(365,195,BVa
l     ,7 );
                if MouseIn(359,209,371,221)then OPTCLIC(365,215,BCi
rc    ,8 );
                if MouseIn(359,229,371,241)then OPTCLIC(365,235,BSe
ta    ,9 );
                if MouseIn(359,249,371,261)then OPTCLIC(365,255,BAr
c     ,10);
                if MouseIn(454,219,466,231)then OPTCLIC(460,225,BEs
tr    ,11);
                if MouseIn(140,285,218,305)then
                   begin
                   Barra  (140,285,2,'APLICAR',78,0);  Car:=#27 ;
                   if Modify then NewLoad:=True;
                   JA:=False;
                   end;
                 if MouseIn(250,285,290,305)then
                   begin
                   Barra  (250,285,2,   'OK'  ,40,0);  Car:=#27 ;
                   end;
                 if MouseIn(360,285,430,305)then
                   begin
                   Barra  (360,285,2,'RELOAD ',70,0);
                   if ArqFixo<>'' then NewLoad:=True;  Car:=#27 ;
                   end;
                 MouseCursor(On );
                 repeat until button=0;
                 Entrou:=True;
                 delay(100);
                 end;
               until (Car=#27)or(button=2);
      END;

  END.
```

```
UNIT JAN345;
{$I-}
INTERFACE

USES CRT,GRAPH,MOUSE,GERAL;

CONST NJHlp=6; {Numero de janelas do menu Help}

VAR JanHlp:integer;{Numero da janela atual do menu Help}

PROCEDURE HELP;
PROCEDURE ABOUT;
PROCEDURE EXIT_;

IMPLEMENTATION

{ }PROCEDURE HELP;
     var Hx,Hy:integer;
     procedure escreve(op:integer;st1,st2:string);
           begin
           SetColor(4);
           OutTextXY(Hx,Hy,st1);
           SetColor(1);
           OutTextXY(Hx+8*length(st1),Hy,st2);
           Hy:=Hy+15*op;
           end;
     BEGIN
     JanHlp:=1;
     repeat MouseCursor(Off);
            Bar(110,18,430,290);
            i :=110; j :=55;
            Hx:=i  ; Hy:=j ;
            SetColor (0);str(JanHlp,aux);
            OutTextXY(240,30,'Pag. '+aux);
            SetColor (8);
            if JanHlp<NJHlp then
               begin
               Line(375,30,425,30); Line(375,31,425,31);
               Line(425,30,415,25); Line(425,31,415,26);
               Line(425,30,415,35); Line(425,31,415,36);
               OutTextXY(375,22,'Enter');
               end;
             if JanHlp>1 then
               begin
               Line(110,30,163,30); Line(110,31,163,31);
               Line(110,30,120,25); Line(110,31,120,26);
               Line(110,30,120,35); Line(110,31,120,36);
               OutTextXY(125,22,'Back ');
```

```
                OutTextXY(125,32,'Space');
                end;
             SetColor(4);
             if JanHlp=1 then
                begin
Escreve  (2,'**',' EXPLICACOES GERAIS:');
Escreve  (1,'*','Esse Programa foi desenvolvido para ma-');
Escreve  (1,'','nipula'+cd1+'ao de Grafos em geral, sendo pos-');
Escreve  (1,'','sivel sua visualiza'+cd1+'ao bem como a execu-');
Escreve  (1,'',cd1+'ao dos algoritmos mais importantes.');
Escreve  (1,'*','O grafo '+ee+' entrado por um arquivo texto ');
Escreve  (1,'','onde cada linha identifica um N'+oo+', infor-');
Escreve  (1,'','mando sua posi'+cd1+'ao e os N'+oo+'s para os quai
s');
Escreve  (2,'','ele aponta seguido do valor do arco.');
Escreve  (2,'*','Os arquivos sao de 3 Tipos: 1,2 e 3');
Escreve  (1,'Tipo 1:','Pode se omitir a posi'+cd1+'ao do N'+oo+'.'
);
Escreve  (1,'','Nesse caso, o programa escolhe uma posi-');
Escreve  (1,'',cd1+'ao adequada para os N'+oo+'s na Tela.');
                end;
             if JanHlp=2 then
                begin
Escreve  (1,'Tipo 2:','Nao pode se omitir nada. '+ee2+' neces-');
Escreve  (1,'','s'+aa+'rio informar a posi'+cd1+'ao do N'+oo+', be
m como');
Escreve  (1,'','aqueles a quem ele liga seguido do valor');
Escreve  (2,'','do respectivo arco.');
Escreve  (1,'Tipo 3:','Pode se omitir os valores dos ar-');
Escreve  (1,'','cos, sendo necess'+aa+'rio apenas a lista dos');
Escreve  (1,'','n'+oo+'s a quem ele liga. O valor do arco tem');
Escreve  (1,'','entao a dist'+aa1+'ncia entre os N'+oo+'s. Caso im
-');
Escreve  (1,'','portante para a situa'+cd1+'ao de localidades,');
Escreve  (2,'','cidades, rios e etc...');
Escreve  (1,'*','O formato do arquivo texto que contem o');
Escreve  (1,'','grafo deve indicar seu tipo na primeira ');
Escreve  (1,'','linha com as palavras:');
Escreve  (1,'','         TIPO1 ou TIPO2 ou TIPO3        ');
                end;
             if JanHlp=3 then
                begin
Escreve  (1,'*','As proximas linhas a partir da segunda ');
Escreve  (1,'','representam os Nos. Caso o arquivo seja ');
Escreve  (1,'','do tipo 2 ou 3, em cada linha deve haver');
Escreve  (1,'','dois numeros iniciais que indicam as po-');
Escreve  (1,'','sicoes X e Y de cada No (separados por  ');
Escreve  (1,'','espaco).');
```

```
Escreve  (1,'*','Em seguida deve constar a lista dos Nos');
Escreve  (1,'','a quem o No em questao se liga, seguido ');
Escreve  (1,'','do respectivo valor do arco caso seja do');
Escreve  (1,'','Tipo 1 ou 2. No final de cada linha deve');
Escreve  (1,'','haver o numero 0(Zero). Pelo tamanho do ');
Escreve  (1,'','arquivo, o programa sabe o numero de Nos');
              end;
            if JanHlp=4 then
              begin
Escreve  (1,'Exemplos:','');
Escreve  (1,'',' TIPO1                    TIPO3          ');
Escreve  (1,'',' 3 12 4 17 0              100 100 3 4 0  ');
Escreve  (1,'',' 1 13 3 7 0               150 230 1 3 0  ');
Escreve  (1,'',' 4 29 2 10 0              200 170 4 2 0  ');
Escreve  (2,'',' 1 18 0                   180 100 1 0    ');
Escreve  (1,'',' TIPO2');
Escreve  (1,'',' 100 100 3 12 4 17 0 ');
Escreve  (1,'',' 150 230 1 13 3 7 0  ');
Escreve  (1,'',' 200 170 4 29 2 10 0 ');
Escreve  (2,'',' 180 100 1 18 0      ');
Escreve  (1,'OBS:',' Todos eles indicam grafos de 4 Nos ');
Escreve  (1,'','muito parecidos.');
               end;
             if JanHlp=5 then
               begin
Escreve  (2,'**',' COMO USAR O PROGRAMA:');
Escreve  (1,'1)','Para fechar uma janela qualquer, basta');
 Escreve (1,'','teclar ESC ou clicar com o botao direito');
 Escreve (1,'','do Mouse. Caso seja um menu de pergunta,');
 Escreve (1,'','somente a tecla ESC finaliza a opera'+ cd1 +'ao.')
;
Escreve  (1,'2)','Clique em FILE para:'                    );
 Escreve (1,'  a)','Criar um Grafo novo.'                   );
 Escreve (1,'  b)','Carregar um Grafo externo.'            );
 Escreve (1,'  c)','Alterar um Grafo j'+ aa +' carregado.'       )
;
 Escreve (1,'  d)','Salvar as modifica‡oes feitas.'        );
Escreve  (1,'3)','Clique em OPTIONS para:'                 );
 Escreve (1,'  a)','Executar algum algoritmo.'            );
 Escreve (1,'  b)','Escolher forma de desenho do grafo.' );
 Escreve (1,'  c)','Recarregar o Grafo com modifica‡oes.');
               end;
             if JanHlp=6 then
               begin
Escreve  (2,'**',' DETALHES:');
Escreve  (1,'*','No desenho do grafo, se este for orien-');
Escreve  (1,'','tado, o valor de cada arco sempre esta a');
Escreve  (1,'','1/4 do No de destino. Caso contrario, se');
```

```
Escreve  (1,'','nao for orientado, o valor fica a 1/2 do');
Escreve  (1,'','No de destino.');
Escreve  (1,'*','Ao excluir um No(File-Edit), os numeros');
Escreve  (1,'','de cada No sao remanejados. Por exemplo:');
Escreve  (1,'','Se excluo o No 4, o no 5 se torna 4,o 6 ');
Escreve  (1,'','se torna 5, 7 vira 6 e assim por diante.');
            end;
         MouseCursor(On);
         SetColor(0); Car:=' ';
         repeat if keypressed then Car:=readkey
                until (Ord(Car) in[8,13,27])or(button=2);
         if (Car=#8 )and(JanHlp>  1  )then JanHlp:=JanHlp-1;
         if (Car=#13)and(JanHlp<NJHlp)then JanHlp:=JanHlp+1;
         until not(Ord(Car)in[8,13]);

   END;
{  }PROCEDURE ABOUT;
   begin
   SetColor(4); Car:=' ';
   OutTextXY ( 130 , 120 , '    INPE/LAC                    ' ) ;
   OutTextXY ( 130 , 140 , 'Bolsa PIBIC/CNPQ                ' ) ;
   SetColor(14);
   OutTextXY ( 130 , 170 , '   Orientador:                  ' ) ;
   OutTextXY ( 130 , 220 , '    Bolsista:                   ' ) ;
   SetColor(15);
   OutTextXY ( 112 , 190 , 'Hor'+aa+'cio Hideki Yanasse ' ) ;
   OutTextXY ( 112 , 240 , 'Rudini Menezes Sampaio      ' ) ;
   SetColor(0);
   OutTextXY ( 110 , 280 , 'Ago/96-97                        ' ) ;
   repeat if keypressed then Car:=readkey
          until (Car=#27)or(button=2);
   end;
{  }PROCEDURE EXIT_;
   Var a:shortint;
   BEGIN
   a:=0;
   SetColor(0);
   OutTextXY(130,110,'Tem certeza que quer');
   OutTextXY(130,120,' sair do programa ?');
   Barra(130,148,2,' IM',0,0);
   Barra(215,148,2,' AO',0,0); SetColor(14);
   OutTextXY(140,155,'S'); OutTextXY(225,155,'N');
   repeat if keypressed then Car:=upcase(readkey);
          if button=1 then
             begin
             if MouseIn(130,148,180,178) then a:=10;
             if MouseIn(215,148,265,178) then a:=01;
             end;
```

```
            until (a<>0)or(button=2)or(Car='S')or(Car='N');
      if (a=10)or(Car='S') then Finalizar(0);
      Car:=' ';
      END;

END.
```

```
UNIT DESENHA;
{$I-}
INTERFACE

USES CRT,GRAPH,MOUSE,GERAL;

PROCEDURE DSN_ARC_ENCAD;

IMPLEMENTATION

{******** Desenha Grafo, Determina os Valores dos Arcos **********
}
{******** E Faz o Encadeamento da Lista de Adjacencias  **********
}
PROCEDURE DSN_ARC_ENCAD;
  {type Pont        = ^CelAdj;
       CelAdj      = Record
                     Vertice:integer;
                     Arco:integer;
                     Prox:pont;
                     End;
       CelulaVert = Record
                     InfoVert:integer;
                     Lista:pont;
                     End;
       Grafo       = Record
                     Espvert:Array[1..Tam]of CelulaVert;
                     NVert:integer;
                     End;}
   var alfa                    :real    ;
       arq                     :text    ;
      {grf                     :Grafo   ;
       p                       :Pont    ;}
       Xp,Xg,Yp,Yg,err,C1,C2:integer ;
       XIgual,Yigual,Meio    :Boolean ;
   begin
   if ArqFixo<>ArqAnt then copia(ArqFixo,arquivo);{Para poder recar
 regar}
   assign(arq,arquivo);
   reset(arq); Xigual:=False; Yigual:=False;
   N:=0; Car:=' ';
   repeat N:=N+1;{N-->Numero de Nos do grafo}
          readln(arq);
          until eof(arq);
   N:=N-1;                  {Tipo 1 --> NOS no circulo            }
   {grf.nvert:=N;}          {Tipo 2 --> NOS em locais determinados}
   close(arq);              {Tipo 3 --> NOS em locais determinados}
   reset(arq);              {           e arcos sao as distancias  }
```

```
readln(arq,aux1);
if (length(aux1)>4)and(copy(aux1,1,4)='TIPO')then
   begin     {AQUI SE VERIFICA O TIPO DO ARQUIVO}
   val(aux1[5],Tipo,err);
   if (err<>0)or(Tipo>3)or(Tipo=0) then N:=0;
   end
else N:=0;
if N=0 then
   begin
   Grava_Cola(10);  Beep;
   barra(100,100,3,'Arquivo em Formato Incorreto',250,40);
   Mwindow(100,100,350,140); MouseCursor(On);
   repeat if keypressed then Car:=readkey
          until (Car=#27)or(button=2)       ;
   MouseCursor(Off); Mwindow(0,0,640,480)   ;
   Grava_Cola(01)  ;
   ArqFixo:=ArqAnt ; exit
   end;
 Apresentacao;
 ESC_NOME_ARQ;
 OutTextXY(55,395,aux1[5]); {aux1[5] , o caracter Tipo}
 ArqAnt:=ArqFixo; JA:=False;
 NFntAnt:=0; NDstAnt:=0;
 ArcGigI:=0;
 if BVerGrf then
    begin
    if Tipo=1 then
       begin
       alfa:=2*PI/N;
       x[1]:=round(Xo+Ro);
       y[1]:=round(Yo);
       for i:=1 to N-1 do
           begin
           x[i+1]:=round(Xo+Ro*cos(i*alfa));
           y[i+1]:=round(Yo-Ro*sin(i*alfa));
           end;
       end;
     if (Tipo in [2,3]) then
       begin
       Xg:=0; Xp:=Gig;
       Yg:=0; Yp:=Gig;
       for i:=1 to N do
           begin
           readln(arq,x[i],y[i]);
           j:=x[i]; k:=y[i];
           if Xg<j then Xg:=j;
           if Xp>j then Xp:=j;
           if Yg<k then Yg:=k;
```

```
            if Yp>k then Yp:=k;
            end;
        if Xg<>Xp then FatorX:=(620-100)/(Xg-Xp) else XIgual:=True
;
        if Yg<>Yp then FatorY:=(460-20 )/(Yg-Yp) else YIgual:=True
;
        for i:=1 to N do
            begin
            if XIgual then x[i]:=360
            else x[i]:=    round(100+(x[i]-Xp)*FatorX);
            if Yigual then y[i]:=240
            else y[i]:=480-round( 20+(y[i]-Yp)*FatorY);{}
            end;
         end;
   end;
   Close(arq); Reset(arq);
   ReadLn(arq,aux1);
   SetColor(CCirc);
   if (BCirc)and(BVerGrf) then
      for i:=1 to N do
          begin
          Circle(x[i],y[i],r1);
          Str(i,aux);
          if i<10 then k:=2 else k:=7;
          OutTextXY(x[i]-k,y[i]-3,aux);
          end;
    SetColor(CArc); C2:=0;
    EulerTour:=True;
    EulerPath:=True;
    for i:=1 to N do for j:=1 to N do C[i,j]:=Gig;
    for i:=1 to N do
        begin
        {grf.espvert[i].infovert:=i;
        new(grf.espvert[i].lista);
        new(p);}
        if (Tipo in [2,3]) then begin read(arq,j);read(arq,j) end;
        read(arq,j); C1:=0;
        if  ( j<>0 ) then
              begin
              {grf.espvert[i].lista^.vertice:=j;
              p:=grf.espvert[i].lista;}
              end;
        while j<>0 do
              begin
              {new(p^.prox);
              p:=p^.prox;
              p^.vertice:=j;}
              C1:=C1+1;
```

```
            if (Tipo=3) then k:=dist(i,j) else read(arq,k);
            {p^.arco:=k;}
            C[i,j]:=k;{Da' o Valor do Arco de I para J}
            if (k<>Gig)and(k>ArcGigI) then ArcGigI:=k;
            Read(arq,j);
            end;
      {if j=0 then grf.espvert[i].lista:=nil else p^.prox:=nil;}
      readln(arq,labels[i]);
      if (Odd(C1)) or  (C1=0)     then EulerTour:=False;{Odd => Imp
ar}
      if                (C1=0)    then EulerPath:=False;
      if (Odd(C1))and(EulerPath) then C2:=C2+1;
      if          not(EulerPath) then C2:=0;
      end;
  if C2<>2 then EulerPath:=False;
  Close(arq);
  i:=1; j:=1;
  SetColor(4);
  Orientad:=False; Meio:=True;
  repeat repeat if (Meio)and(C[i,j]<>Gig)and(C[j,i]=Gig)then Meio:
=False;
                if not(Meio or Orientad)then Orientad:=True;
                if not(Orientad)and(C[i,j]<>C[j,i])then Orientad:=
True;
                j:=j+1;
                until (Orientad)or(j>N);
         j:=1;
         i:=i+1;
         until (Orientad)or(i>N);
   EulerTour:=EulerTour and Meio;{Meio aqui define se os arcos vao
  e }
   EulerPath:=EulerPath and Meio;{voltam, mesmo com valores diferen
 tes}
   SetColor(CArc);
   Meio:=((Tipo=3)or(not(Orientad)));{Meio aqui define se escreve o
  }
   for i:=1 to N do                    {valor do arco no meio ou se a
  }
       for j:=1 to N do                { 1/4 do no de destino
  }
         begin
         if (not(Orientad)and(i>j)) then k:=Gig else k:=C[i,j] ;
         if k<>Gig then
            begin
            if (BArc)and(BVerGrf) then liga(i,j);
            Str(k,aux);
            SetColor(CCust);
            if (BVal)and(BVerGrf) then
```

```
                if Meio then OutTextXY(round((x[i] + x[j])/2)-3,
                                       round((y[i] + y[j])/2)-7,au
x)
                else         OutTextXY(round((x[i]+3*x[j])/4)-3,
                                       round((y[i]+3*y[j])/4)-7,au
x);
              SetColor(CArc);
              end;
            end;
  SetColor(14);
  if not(BCirc)and(BVerGrf) then for i:=1 to N do PieSlice(x[i],y[
i],0,360,r2);
  Str(ArcGigI,ArcGigS);
  end;


END.
```

```
UNIT IMPLEMEN;
{$I-}
INTERFACE

USES CRT,GRAPH,MOUSE,GERAL,ALGORIT;

PROCEDURE VERCARACT     ;
PROCEDURE MENOR_CAMINHO;
PROCEDURE SPANNING_TREE;

IMPLEMENTATION

PROCEDURE OUT_OK;
   begin
   Grava_Cola(01);
   MouseCursor(On);
   end;
PROCEDURE OUT_ERRO;
   begin
   Beep;
   Grava_Cola(01);
   Grava_Cola(10);
   SetColor(14);
   Barra(100,100,3,' Opcao Errada ',140,30);
   MWindow(100,100,240,130);
   MouseCursor(On);
   repeat if keypressed then car:=readkey;
          until (car=#27)or(button=2);
   MouseCursor(Off)    ;
   Grava_Cola (01 )    ;
   MouseCursor(On )    ;
   MWindow(0,0,640,480);
   end;
PROCEDURE VERCARACT;{Ver Caracteristicas do Grafo}
     var Cnx:Shortint;
     begin
     MouseCursor(Off);
     Grava_Cola (10 );
     Cnx:=Conexo;
     EulerTour:=EulerTour and (Cnx<>0);
     EulerPath:=EulerPath and (Cnx<>0);
     SetColor(1);
     Barra(215,70,7,'',295,235);
     Rectangle (220,75,505,300);
     MWindow   (220,75,505,300);
     SetColor(8); Car:=' ';
     Line(410,75,410,300);
     for i:=0 to 5 do Line(220,75+25*i,505,75+25*i);
```

```
    SetColor(1);
    if    Orientad     then aux:='SIM' else aux:='NAO';
    OutTextXY(230,85 ,'Orientado/Direcionado    '+aux);
    if     Cnx<> 0     then aux:='SIM' else aux:='NAO';
    OutTextXY(230,110,'       Conexo            '+aux);
    if     Cnx = 2     then aux:='SIM' else aux:='NAO';
    OutTextXY(230,135,' Fortemente Conexo       '+aux);
    if    EulerTour    then aux:='SIM' else aux:='NAO';
    OutTextXY(230,160,'   Tour  Euleriano       '+aux);
    if    EulerPath    then aux:='SIM' else aux:='NAO';
    OutTextXY(230,185,' Caminho Euleriano       '+aux);
    MouseCursor(On);
    repeat if keypressed then Car:=readkey;
           until (Car=#27)or(button=2);
    MouseCursor(Off);
    Grava_Cola (01 );
    MouseCursor(On );
    MWindow(0,0,640,480);
    end;
PROCEDURE Dijkst;{Aplicacao do Algoritmo de Dijkstra}
   Var No_Font,No_Dest,Naux,A: integer  ;
       D,Cam,MenC:  Vetor;
       strin:string;
   Begin
   MouseCursor(Off);
   Grava_Cola(10); SetColor(1);
   Barra(90,92,3,'   Entre com o N£mero do N'+oo+' Fonte ou Raiz:
',390,130);
   SetColor(15); Str(N,strin);
   Barra(260,109,0,'',40,20);
   LerInt(No_Font,strin,270,115,Length(strin));
   if No_Font=0 then begin Out_Ok; exit end;
   if not(No_Font in[1..N]) then begin Out_Erro; exit end;
   SetColor(15) ;
   if BVerGrf then
      begin
      OutTextXY(103,137,'Digite (1) para ver os Valores das Distƒn
cias');
      OutTextXY(162,149,'do N'+oo+' Fonte para os outros N'+oo+'s
    ');
      OutTextXY(273,161,'OU'  );
      OutTextXY(100,173,'Digite (2) s'+oo+' para ver o Caminho par
a algum N'+oo+':');
      Barra(260,188,0,'',40,20);
      LerInt(Naux,strin,270,194,1);
      if Naux=0 then begin Out_Ok; exit end;
      if not(Naux in[1,2]) then begin Out_Erro; exit end;
      end
```

```
else Naux:=1;
Grava_Cola(01);
MouseCursor(On);
DIJKSTRA(No_Font,D,Cam);
if (Naux=1) then
     begin
     MouseCursor(Off);
     Grava_Cola(10);
     MWindow(218,78,507,282);
     MouseCursor(On);
     A:=0; i:=0;
     Car:=#13;
     repeat if (Car=#13)or(button=1) then
                begin
                if i>=N then A:=0;
                MouseCursor(Off);
                Barra(215,75,7,'',295,210);
                Rectangle(218,78,507,282);
                Line(360,78,360,282);
                OutTextXY(380,260,'<ENTER>-NEXT 10');
                SetColor(1);
                Str(No_Font,aux);
                OutTextXY(370,100,'N'+oo+' de Referˆncia');
                OutTextXY(385,115,'para Menores');
                OutTextXY(385,130,'Caminhos:> '+aux);
                SetColor(0);
                for i:=10*A+1 to 10*(A+1) do
                    begin
                    if i<=N then
                       begin
                       k:=(i-1)mod 10;
                       j:=D[i];
                       Str(i,aux);
                       SetColor(8);
                       Line(218,102+20*k,360,102+20*k);
                       SetColor(1);
                       OutTextXY(231,90+20*k,aux);
                       Line(252,93+20*k,269,93+20*k);
                       OutTextXY(240,90+20*k,'   >');
                       if j=Gig then
                          if i=No_Font then aux:='0'
                          else aux:=#236{oo=INFINITO}
                       else str(j,aux);
                       OutTextXY(278,90+20*k,aux);
                       end;
                    end;
               A:=A+1; Car:=' ';
               MouseCursor(On) ;
```

```
                end;
              if keypressed then Car:=readkey;
              until (Car=#27)or(button=2);
        MouseCursor(Off);
        Grava_Cola(01);
        MouseCursor(On);
        MWindow(0,0,640,480);
        end;
    if (Naux=2) then
        begin
        MouseCursor(Off); Grava_Cola(10);
        SetColor(1); Str(N,strin);
        Barra(90,92,3, 'Entre com o N£mero do N'+oo+' Destino:',290
,60);
        SetColor(15);   Barra(220,109,0,'',30,20);
        LerInt(No_Dest,strin,230,115,Length(strin));
        if No_Dest=0 then begin Out_Ok; exit end;
        if not(No_Dest in[1..N])or(No_Dest=No_Font) then
           begin Out_Erro; exit end;
        Grava_Cola(01); MouseCursor(On);
        SetLineStyle(3,0,0);
        Str(No_Dest,aux);
        SetColor(14);
        for i:=1 to N do MenC[i]:=No_Dest;{MenC marca o caminho exa
to para}
        j:=1;{ir de NoFont a NoDest. Se o primeiro elemento for 0,
nao existe}
        k:=No_Dest;{tal caminho}
        repeat i:=Cam[k];
               if not(i in[0,No_Font]) then begin MenC[j]:=i; k:=i
end;
               j:=j+1;
               until (i in[0,No_Font]);
        for k:=1 to j-2 do xyapg[k]:=MenC[j-k-1];
        for k:=1 to j-2 do MenC[k]:=xyapg[k];
        if i=0 then MenC[1]:=0;
        for i:=1 to N do XYapg[i]:=MenC[i];
        NFntAnt:=No_Font;{Servem para riscar por cima quando}
        NDstAnt:=No_Dest;{o algoritmo for chamado novamente }
        InterLiga(MenC,No_Font,No_Dest,01);
        end;
    SetLineStyle(0,0,1);
    end;
 {***************** FLOYD ********************}
 PROCEDURE Floy;{Aplicacao do algoritmo de Floyd}
    var A,B,Step,OrdMat:integer;
        D,Cam:Matriz;
        ParaEntrar,MaisQStr:Boolean;
```

```
  Begin
  FLOYD(D,Cam);
  MouseCursor(Off);
  Grava_Cola(10);
  A:=0; i:=0; j:=0; Step:=8*length(ArcGigS)+10;
  B:=0; ParaEntrar:=True; Car:=' ';
  if Step=18 then Step:=25 ;
  if N>10 then OrdMat:=10 else OrdMat:=N;
  if Step*OrdMat>190 then MaisQStr:=True else MaisQStr:=False
;
  if MaisQStr then MWindow(90,25,130+Step*OrdMat,115+20*OrdMat)
  else             MWindow(90,25,      320      ,115+20*OrdMat)
;
  MouseCursor(On);
  Repeat if (ord(Car) in[72,75,77,80])or(ParaEntrar) then
            begin ParaEntrar:=False;
            MouseCursor(Off);
            if MaisQStr then
                 Barra(85,20,7,'', 50+Step*OrdMat,100+20*OrdMat)
            else Barra(85,20,7,'',     240       ,100+20*OrdMat)
;
            SetColor(1);
            if MaisQStr then
                 Rectangle(90,25,130+Step*OrdMat,115+20*OrdMat)
            else Rectangle(90,25,      320      ,115+20*OrdMat)
;           OutTextXY(100,    40      ,'Matriz de Menores Caminho
s');
            OutTextXY(100,85+20*OrdMat,'As Setas do Teclado movem
 ');
            OutTextXY(100,95+20*OrdMat,' a Matriz Caso Grau > 10
 ');
            SetColor(8);
            for i:=0 to OrdMat do
                line(115 , 70+20*i,115+Step*OrdMat,70+20*i);
            for i:=0 to OrdMat do
                line(115+Step*i,70,115+Step*i,70+20*OrdMat);
            for i:=10*A+1 to 10*(A+1) do
                begin
                if i<=N then
                   begin
                   k:=(i-1)mod 10;
                   Str(i,aux);
                   OutMeioXY(95,70+20*k,20,aux);
                   end;
                end;
             for i:=10*B+1 to 10*(B+1) do
                begin
```

```
                if i<=N then
                   begin
                   k:=(i-1)mod 10;
                   Str(i,aux);
                   OutMeioXY(115+Step*k,53,Step,aux);
                   end;
                end;
             SetColor(1);
             for i:=10*A+1 to 10*(A+1) do
                begin
                k:=(i-1)mod 10;
                for j:=10*B+1 to 10*(B+1) do
                   if (i<=N)and(j<=N) then
                      begin
                      l:=(j-1)mod 10;
                      if D[i,j]=Gig then
                         begin
                         aux:=#236 { oo };
                         SetColor(0);
                         end
                       else Str(D[i,j],aux);
                       if i=j then SetColor(4);
                       OutMeioXY(115+Step*l,70+20*k,Step,aux);
                       SetColor(1)
                       end;
                 end;{ Nao usar I nem J ate depois do Until }
              SetColor(0);
              Car:=' ' ; MouseCursor(On);
              end;
           if keypressed then Car:=readkey;
           if (Car=#00) then Car:=readkey;
           if (Car=#72) then if(A>0) then A:=A-1
                             else Car:=' '{A:=trunc(N/10){} ;
           if (Car=#80) then if(i<N) then A:=A+1
                             else Car:=' '{A:= 0{}          ;
           if (Car=#75) then if(B>0) then B:=B-1
                             else Car:=' '{B:=trunc(N/10){} ;
           if (Car=#77) then if(j<N) then B:=B+1
                             else Car:=' '{B:= 0{}          ;
           {if (ord(Car) in[72,75,77,80])and(N<=10) then Car:=' ';{
}
           until (Car=#27) or (button=2);{Os 4 Coments acima, se}
                                         {forem ser executados, }
                                         {  devem o ser juntos  }
   MouseCursor(Off);
   Grava_Cola(01)  ;
   MouseCursor(On) ;
   MWindow(0,0,640,480);
   End;
{********************** Menor Caminho ****************************
```

```
}
PROCEDURE MENOR_CAMINHO;
   begin
   if Dijk  then begin TipAlg:=1; Dijkst end;
   if FLd   then begin TipAlg:=2; Floy   end;
   end;
{********************* Spanning Tree ******************************
}
PROCEDURE SPANNING_TREE;{Aplicacao do algoritmo de Prim}
   begin
   TipAlg :=3;                    {          1 : Dijkstra       }
   NFntAnt:=0;                    {          2 : Floyd          }
   SetColor(CT);                  {          3 : Spanning Tree  }
   SetLineStyle(3,0,0);
   MouseCursor(Off);
   PRIM;
   SetLineStyle(0,0,1);
   MouseCursor(On);
   end;
END.
```

```
UNIT ALGORIT;
{$I-}
INTERFACE

USES CRT,GRAPH,MOUSE,GERAL;

FUNCTION  CONEXO: ShortInt;
PROCEDURE DIJKSTRA(NoF:integer;var D:Vetor;var Cam:Vetor);
PROCEDURE FLOYD(var D:Matriz;var Cam:Matriz);
PROCEDURE PRIM;

IMPLEMENTATION

FUNCTION CONEXO: ShortInt;{ CONEXO=2  =>  FORTEMENTE CONEXO
   }
   var D,Cam:Matriz;       { CONEXO=1  =>  SIMPLESMENTE CONEXO
   }
       V,S:conjunto;       { CONEXO=0  =>  NAO CONEXO
   }
       FortConex:Boolean;
   BEGIN              { Por Conexo pode se entender, Conectado.
   }
   Floyd(D,Cam);      {     Fortemente Conexo quer dizer que de tod
os }
   V:=[1..N];         { os Nos podemos chegar em todos os outros.
   }
   S:=[2..N];         {     Simplesmente Conexo quer dizer que, se
o  }
   Conexo:=2;         { Grafo fosse nao-direcionado, ele seria fort
e- }
   FortConex:=True;   { mente conexo.
   }
   for i:=1 to N do   {      Nao Conexo e' quando nao e' nenhum aci
ma.}
       for j:=1 to N do
           if D[i,j]=Gig then FortConex:=False;
   if FortConex then exit;
   Conexo:=1;
   for i:=1 to N do
       begin
       if (i in S) then
           for j:=1 to N do
               if (j in (V-S)) then
                   if D[i,j]<>Gig then S:=S-[i];
       if (i in (V-S)) then
           for j:=1 to N do
               if (j in S) then
                   if D[i,j]<>Gig then S:=S-[j];
```

```
      end;
   if S<>[] then Conexo:=0;
   END;
{****************** DIJKSTRA ********************}
PROCEDURE DIJKSTRA(NoF:integer;var D:Vetor;var Cam:Vetor);
   Var S,V: conjunto ; {ALGORITMO DE DIJKSTRA}
       Biz: integer  ;
   Begin
   S:=[NoF] ;{Contem os Nos cuja minima distancia a No_Font  }
             {           ja' esta' determinada               }
   V:=[1..N];{            Contem Todos os Nos                }
   for i:=1 to N do Cam[i]:=NoF;
   Cam[NoF]:=0;{Marca o No que se Liga ao Indice, para se ir ate'
No_Font}
   for i:=1 to N do D[i]:=C[NoF,i];
   for i:=1 to N-1 do      { Exemplo: X2.PAS , para o No' 5 }
       begin               { Cam = [5,3,1,2,0,0] }
       Biz:=NoF;           { 0 para Dizer que nao ha' Caminho }
       for j:=1 to N do
           if (j in (V-S)) then
              if D[j]<=D[Biz] then
                 Biz:=j;
       S:=S+[Biz];                          { C[A,A]=oo }
       for j:=1 to N do
           if ((j in (V-S))and(C[Biz,j]<Gig)and(D[Biz]<Gig)) then
              begin
              D[j]:=min(D[j],D[Biz]+C[Biz,j]);
              if D[j]=D[Biz]+C[Biz,j] then Cam[j]:=Biz;
              end;
       end;
    for i:=1 to N do
       if (D[i]=Gig)and(i<>NoF) then Cam[i]:=0;
    end;
 {****************** FLOYD ********************}
 PROCEDURE FLOYD(var D:Matriz;var Cam:Matriz);
    Var i,j,k,Dij:integer;{ALGORITMO DE FLOYD}
    Begin
    for i:=1 to N do
        for j:=1 to N do
            begin
            if i=j  then D[i,j]:=0
            else D[i,j]:=C[i,j];
            if i=j  then Cam[i,j]:=0
            else Cam[i,j]:=  i   ;
            end;
     for k:=1 to N do
         for i:=1 to N do
             for j:=1 to N do
```

```
                begin
                Dij:=D[i,j];
                if (D[i,k]<>Gig)and(D[k,j]<>Gig) then
                    D[i,j]:=min(D[i,j],D[i,k]+D[k,j]);
                if (D[i,j]<>Dij) then Cam[i,j]:=Cam[k,j] ;
                end;
   End;
{******************* PRIM *********************}
PROCEDURE PRIM;{ALGORITMO DE PRIM}
   var S,V:conjunto;
       NAt1,NAt2,Min,DNoAt:integer;
   begin
   V:=[1..N];
   S:=[1];
   NAt1:=1;
   NAt2:=1;
   while (S<>V) do
         begin
         i:=N;
         Min:=Gig;
         while ((i<>0)and(S<>V)) do
               begin
               if (i in S) then
                  begin
                  for j:=N downto 1 do
                      begin
                      DNoAt:=C[i,j];
                      if (j in(V-S)) then
                         if DNoAt<=Min then
                            begin
                            Min:=DNoAt;
                            NAt1:=i;
                            NAt2:=j;
                            end;
                      end;
                  end;
               i:=i-1;
               end;
          S:=S+[NAt2];
          if Min<>Gig then
             begin
             Liga(NAt1,NAt2);
             if BAtraso then delay(1000);
             end;
          end;
     end;
  END.
```

```
UNIT MOUSE;

INTERFACE

USES CRT,DOS;

VAR regs : registers;

TYPE onoff = (on,off);

FUNCTION initmouse : boolean;
FUNCTION getmousex : integer;
FUNCTION getmousey : integer;
FUNCTION button    : integer;
PROCEDURE mousecursor(com :  onoff );
PROCEDURE setmouse   (x,y : integer);
PROCEDURE mwindow    (x1,y1,x2,y2:integer);

IMPLEMENTATION

FUNCTION PSeg( x : POINTER ) : Word; INLINE( $5a/$58 );
FUNCTION POfs( x : POINTER ) : Word; INLINE( $58/$5a );


function initmouse: boolean;
   begin
   regs.ax:=$00;
   intr($33,regs);
   if regs.ax=0 then
      initmouse := false;
   if regs.ax=-1 then
      initmouse := true;
   end;

procedure mousecursor(com : onoff);
   begin
   if com = off then
      begin
      regs.ax := $02;
      intr($33,regs);
      end;
   if com = on then
      begin
      regs.ax := $01;
      intr($33,regs);
      end;
   end;
```

```
function getmousex : integer;
   begin
   regs.ax := $03;
   intr($33,regs);
   getmousex := regs.cx;
   end;

function getmousey : integer;
   begin
   regs.ax := $03;
   intr($33,regs);
   getmousey := regs.dx;
   end;

procedure setmouse(x,y : integer);
   begin
   regs.ax := $04;
   regs.dx := y;
   regs.cx := x;
   intr($33,regs);
   end;

function button : integer;
   begin
   regs.ax := $03;
   intr($33,regs);
   button := regs.bx;
   end;

procedure mwindow(x1,y1,x2,y2 : integer);
   begin
   regs.ax := $08;
   regs.cx := y1;
   regs.dx := y2;
   intr($33,regs);

   regs.ax := $07;
   regs.cx := x1;
   regs.dx := x2;
   intr($33,regs);
   end;

END.
```

```
UNIT GERAL;
{$I-}
INTERFACE

USES CRT,GRAPH,DOS,MOUSE;

CONST PI  = 3.14159265359  ;
      Gig = 32767          ;
      Tam = 100            ;
      r1  = 9              ;
      r2  = 3              ;
      Xo  = 360            ;
      Yo  = 240            ;
      Ro  = Yo-r1-2        ;

      aa  = #160           ;            {   a      agudo
}
      ee  = #130           ;            {   e       "
}
      ii  = #161           ;            {   i       "
}
      oo  = #162           ;            {   o       "
}
      uu  = #163           ;            {   u       "
}
      aa1 = #131           ;            {   a    circunflexo
}
      ee1 = #136           ;            {   e       "
}
      oo1 = #147           ;            {   o       "
}
      aa2 = #133           ;            {   a     grave
}
      ee2 = #144           ;            {   E     agudo
}
      cd1 = #135           ;            {   c     cedilha
}
      cd2 = #128           ;            {   C     cedilha
}
      grau= #248           ;            {         grau
}

TYPE Conjunto = Set of Byte                    ;
     Vetor = array[1..Tam]of integer           ;
     Matriz = array[1..Tam,1..Tam]of integer;

VAR BLabel,BCarac,Algor     :Boolean                    ;
    MenorCam,SpanTr         :Boolean                    ;
```

```
    Dijk,FLd                    :Boolean                          ;
    BVerGrf,BMouse,BAtraso      :Boolean                          ;
    BVal,BCirc,BEstr,BSeta      :Boolean                          ;
    BArc,EulerTour,EulerPath:Boolean                              ;
    Modify,Orientad,Reload      :Boolean                          ;
    NewLoad,JA                  :Boolean                          ;
    CArc,CCust,CCirc,CMen       :integer                          ;
    GD,GM,CF,CT,Tipo,N          :integer                          ;
    i,j,k,l,TipAlg,ArcGigI      :integer                          ;
    FatorX,FatorY               :real                             ;
    arquivo,ArqAnt,ArqFixo      :String                           ;
    x,y                         :Vetor                            ;
    labels                      :Array[1..Tam]of string[18]       ;
    aux,ArcGigS                 :String[5]                        ;
    C                           :Matriz                           ;
    SS                          :PathStr                          ;
    PP1,PP2,PP3                 :Pointer                          ;
    vXmouse,vYmouse             :Integer                          ;
    aux1,aux2,ArqLixo           :String                           ;
    Car                         :Char                             ;
    NFntAnt,NDstAnt             :integer                          ;
    XYapg                       :Vetor                            ;

PROCEDURE Beep
;
PROCEDURE OutMeioXY(x,y,step:integer;strin:string)
;
FUNCTION  MOUSEIN(x1,y1,x2,y2:integer)       :Boolean
;
FUNCTION  MIN(x,y:integer)                   :integer
;
FUNCTION  Dist(i,j:integer)                  :longint
;
FUNCTION  Angulo(x1,y1,x2,y2:integer)        :real
;
PROCEDURE seta(x,y:integer;tt:real)
;
PROCEDURE LIGA(i,j:integer)
;
PROCEDURE BARRA(a,b,d:integer;e:string;f,g:integer)
;
PROCEDURE REBAIXO(Blok:shortint)
;
PROCEDURE GRAVA_COLA(GC:shortint)
;
PROCEDURE COPIA(arq1,arq2:string)
;
PROCEDURE LERINT(var Ai:integer;var As:string;x,y,NumDig:integer)
```

```
;
PROCEDURE LERSTR(var B:string ;x,y,NumDig:integer)
;
PROCEDURE FINALIZAR(Erro:integer)
;
PROCEDURE APRESENTACAO
;
PROCEDURE ESC_NOME_ARQ
;
PROCEDURE SALV_ALTERACOES
;
PROCEDURE ZOOM
;
PROCEDURE VERLABELS
;
PROCEDURE INTERLIGA(A:Vetor;NoF,NoD,ApRisc:integer)
;


IMPLEMENTATION


PROCEDURE Beep;
   begin
   sound(1000);
   delay(100);
   nosound;
   end;
PROCEDURE OutMeioXY(x,y,step:integer;strin:string);
   begin
   OutTextXY(round(x-4*length(strin)+step/2),y+7,aux);
   end;
FUNCTION MOUSEIN(x1,y1,x2,y2:integer):Boolean;
   var x,y:integer;
   begin
   x:=GetMouseX;
   y:=GetMouseY;
   if (x>x1)and(x<x2)and(y>y1)and(y<y2) then MouseIn:=True
   else MouseIn:=False;
   end;
FUNCTION MIN(x,y:integer):integer;
   begin{Devolve o menor numero entre Dois}
   if x>y then min:=y else min:=x;
   end;
FUNCTION Dist(i,j:integer):longint;
   var a,b:longint;
   begin
   a:=round((x[i]-x[j])/FatorX);
```

```
   b:=round((y[i]-y[j])/FatorY);
   a:=sqr(a);
   b:=sqr(b);
   a:=round(sqrt(a+b));
   dist:=a;
   end;
FUNCTION Angulo(x1,y1,x2,y2:integer):real;
   var x,y:integer;{Da o Angulo entre os Pontos (X1,Y1) e (X2,Y2
)}
       tt:real;
   begin
   x:=x2-x1;
   y:=y2-y1;
   if x=0 then
      if y>0 then tt:=PI/2
      else if y<0 then tt:=1.5*PI
           else tt:=0
   else tt:=arctan(y/x);
   if x<0 then Angulo:=PI+tt
   else if y<0 then Angulo:=2*PI+tt else Angulo:=tt;
   end;
FUNCTION pt(x:integer;y:real):integer;
   begin{Auxilio para Ligacao entre Nos nao Entrar nos Circulozinh
os}
   pt:=round(x+r1*y);
   end;
FUNCTION gt(x:integer;y:real):integer;
   begin{Auxilio para Ligacao entre Nos nao Entrar nos Circulozinh
os}
   gt:=round(x+r2*y);
   end;
PROCEDURE seta(x,y:integer;tt:real);
   var i,j:integer;{Desenha a Setinha}
   begin
   for j:=2 to 5 do
       begin
       i:=5*j;
       line(x,y,pt(x,-cos(tt-PI/i)),pt(y,-sin(tt-PI/i)));
       line(x,y,pt(x,-cos(tt+PI/i)),pt(y,-sin(tt+PI/i)));
       end;
   end;
PROCEDURE LIGA(i,j:integer);
   var tt:real;{Liga Dois Pontos definidos nas Matrizes X e Y}
   begin       {Onde I e J sao os Indices}
   tt:=Angulo(x[i],y[i],x[j],y[j]);
   if not(BCirc) then
      if (BEstr)and(Tipo=3) then
          begin SetLineStyle(0,0,1);
```

```
Line(gt(x[i],-sin(tt)),gt(y[i],cos(tt)),gt(x[j],-sin(tt)),gt(y[j],
cos(tt)));
Line(gt(x[i],sin(tt)),gt(y[i],-cos(tt)),gt(x[j],sin(tt)),gt(y[j],-
cos(tt)));
SetLineStyle(3,0,0); line(x[i],y[i],x[j],y[j]); SetLineStyle(0,0,1
);
          end
      else Line(x[i],y[i],x[j],y[j])
   else
Line(pt(x[i],cos(tt)),pt(y[i],sin(tt)),pt(x[j],-cos(tt)),pt(y[j],-
sin(tt)));
   if (BSeta)and(Orientad)then Seta(pt(x[j],-cos(tt)),pt(y[j],-sin
(tt)),tt);
   end;
PROCEDURE BARRA(a,b,d:integer;e:string;f,g:integer);
   var c:shortint;
   begin
   if f=0 then f:=50;
   if g=0 then g:=20;
   SetFillStyle(1,d);
   Bar(a,b,a+f,b+g);
   Rectangle(a,b,a+f,b+g);
   SetColor(15);
   if (length(e)<4)or(length(e)>6) then c:=-2 else c:=0;
   OutTextXY(a+10-c,b+7,e);
   SetColor(0);
   end;
PROCEDURE REBAIXO(Blok:shortint);
   begin
   MouseCursor(Off);
   SetColor(3);
   SetFillStyle(1,0);
   Bar3D(8,50*Blok+3,70,50*Blok+20,5,TopOn);
   SetFillStyle(1,14);
   SetColor(9);
   Bar3D(10,50*Blok+5,71,50*Blok+18,3,TopOn);
     delay(200);
   SetColor(3);
   SetFillStyle(1,0);
   Bar3D(10,50*Blok+5,71,50*Blok+18,3,TopOn);
   SetFillStyle(1,14);
   SetColor(9);
   Bar3D(8,50*Blok+3,70,50*Blok+20,5,TopOn);
   end;
PROCEDURE GRAVA_COLA(GC:shortint);
   var Size:LongInt;
   begin
   Size:=ImageSize(0,0,639,120);{38726}
```

```
  if GC=10 then
     begin
     GetMem  ( PP1 , Size );    { Aloca memoria na area heap }
     GetMem  ( PP2 , Size );
     GetMem  ( PP3 , Size );
     GetImage(  0  ,   0  , 639 , 120 , PP1^ );
     GetImage(  0  , 120  , 639 , 240 , PP2^ );
     GetImage(  0  , 240  , 639 , 320 , PP3^ );
     end;
  if GC=01 then
     begin
     i:=0;{NormalPut} {i pode ser 0,1,2,3 ou 4} {Bizus i=0 e i=4}
     PutImage( 0 ,  0  , PP1^ , i );            {Resto e' igual }
     PutImage( 0 , 120 , PP2^ , i );
     PutImage( 0 , 240 , PP3^ , i );
     FreeMem( PP1, Size);
     FreeMem( PP2, Size);
     FreeMem( PP3, Size);
     Dispose(PP1);Dispose(PP2);Dispose(PP3);
     end;
  end;
PROCEDURE COPIA(arq1,arq2:string);
  var arquivo1,arquivo2:text;
      wrd:string;
  begin
  assign  (arquivo1,arq1);
  assign  (arquivo2,arq2);
  reset   (  arquivo1  ) ;
  rewrite (  arquivo2  ) ;
  while not(eof(arquivo1)) do
        begin
        readln (arquivo1,wrd);
        writeln(arquivo2,wrd);
        end;
  close(arquivo1);
  close(arquivo2);
  end;
PROCEDURE LERINT(var Ai:integer;var As:string;x,y,NumDig:integer);
  var Cor,erro,dx:integer;
      B:string;
  begin
  Cor:=GetColor;
  SetColor(7);
  car:=readkey;
  B:=''; dx:=0;
  while (car<>#13)and(car<>#27) do
        begin
        if (car=#8)and(length(B)>0) then
```

```
            begin
            Setcolor(0); dx:=dx-8;
            OutTextXY(x+dx,y,B[length(B)]);
            B:=copy(B,1,length(B)-1);
            SetColor(7);
            end;
         if ((ord(car)-48)in[0..9])and(length(B)<NumDig) then
            begin
            OutTextXY(x+dx,y,car);
            B:=B+car; dx:=dx+8;
            end;
         car:=readkey;
         end;
   val(B,Ai,erro); As:=B;
   if (car=#27)or(erro<>0) then begin Ai:=0;As:=' ' end;
   SetColor(Cor); Car:=' '; {Se Ai=0 deu ESC ou ERRO}
   end;
PROCEDURE LERSTR(var B:string;x,y,NumDig:integer);
   var Cor,dx,passo:integer;
       aux:char;
   begin
   passo:=8;
   Cor:=GetColor;
   SetColor(7);
   if NumDig<18 then car:=upcase(readkey) else car:=readkey;{ >17
(Label) }
   B:=''; aux:=' '; dx:=0;
   while (car<>#13)and(car<>#27) do
         begin
         if (car=#8)and(length(B)>0) then
            begin
            Setcolor(0); dx:=dx-passo;
            OutTextXY(x+dx,y,B[length(B)]);
            B:=copy(B,1,length(B)-1);
            SetColor(7);
            end;
         if (length(B)<NumDig) and
         not (((ord(car)-48)in[0..9]) and (dx=0)) then
            if ord(car)>31 then
               begin
               OutTextXY(x+dx,y,car);
               B:=B+car; dx:=dx+passo;
               end;
         aux:=car;
         if NumDig<18 then car:=upcase(readkey) else car:=readkey;
         end;                                                    { >17
(Label) }
   if car=#27 then B:=''; {Se B='' deu ESC}
```

```
   SetColor(Cor); Car:=' ';
   end;
PROCEDURE FINALIZAR(Erro:integer);
   begin
   SALV_ALTERACOES ;
   MouseCursor(Off);
   Grava_Cola(01)  ;
   Barra(100,100,3,'O programa esta sendo finalizado',280,40);
   SetColor(15);
   if Erro<>0 then OutTextXY(130,120,'devido a um erro de sistema'
);
   delay(500)  ;
   Mwindow(0,0,640,480);
   CloseGraph  ;
   halt;
   end;
PROCEDURE APRESENTACAO;
   begin
   ClearDevice;
   SetBkColor(CF);
   SetColor(9);
   SetFillStyle(1,CMen);
   Bar(3,4,78,476);
   SetFillStyle(1,14);
   Bar3D( 8 ,53  , 70 ,  70 , 5  , TopOn);{FILE}
   Bar3D( 8 ,103 , 70 , 120 , 5  , TopOn);{OPTIONS}
   Bar3D( 8 ,153 , 70 , 170 , 5  , TopOn);{HELP}
   Bar3D( 8 ,203 , 70 , 220 , 5  , TopOn);{ABOUT}
   Bar3D( 8 ,253 , 70 , 270 , 5  , TopOn);{EXIT}
   OutTextXY( 24 , 59 ,'FILE     ' );
   OutTextXY( 12 , 109,'OPTIONS  ' );
   OutTextXY( 24 , 159,'HELP     ' );
   OutTextXY( 20 , 209,'ABOUT    ' );
   OutTextXY( 24 , 259,'EXIT     ' );
   SetColor(8);{0}
   OutTextXY( 10 , 325,'ARQUIVO: ' );
   OutTextXY( 10 , 360,'T□RMINO: ' );
   OutTextXY( 10 , 395,'TIPO:    ' );
   SetColor(14);
   Rectangle(1 ,1,80 ,479);
   Rectangle(80,1,639,479);
   SetColor(9);
   end;
PROCEDURE ESC_NOME_ARQ;
  var J,ponto:integer;
  begin
  SetColor(4);
  J:=length(ArqFixo);
```

```
 ArqAnt:=ArqFixo;
 ponto:=0;
 for i:=1 to J do
     begin
     if ArqAnt[i]='.' then ponto:=i;
     ArqAnt[i]:=upcase(ArqAnt[i]) ;
     end;
 if ponto=0  then OutTextXY(10,340,copy(ArqAnt,1,8));
 if ponto<>0 then
    begin
    OutTextXY(10,340,copy(ArqAnt,1,ponto-1));
    OutTextXY(10,375,copy(ArqAnt,ponto+1,3));
    ArqAnt:=copy(ArqAnt,1,ponto+3);
    ArqFixo:=ArqAnt;
    end;
 end;
PROCEDURE SALV_ALTERACOES;
   var OptC:char;
       OptS:string;
   begin
   if Modify then
      begin
      MouseCursor(Off)     ;
      Grava_Cola(01)        ;
      Grava_Cola(10)         ;
      SetColor(1)             ;
      SetFillStyle(1,7)        ;
      Bar(210,50,465,155)       ;
      Rectangle(215,55,460,150) ;
      OutTextXY(230,65,'Deseja salvar as alteracoes');
      OutTextXY(230,80,'  do arquivo '+ArqFixo+' ? ');
      OutTextXY(350,107,'(S/N)')    ;
      SetFillStyle(1,0)              ;
      repeat Beep                     ;
             Bar(300,100,330,120)      ;
             LerStr(OptS,312,107,1)     ;
             OptC:=UpCase(OptS[1])       ;
             until ((OptC='S')or(OptC='N')or(OptC=''));
      if OptC='S' then Copia(arquivo,ArqFixo);
      Grava_Cola(01);
      Grava_Cola(10);
      MouseCursor(On);
      Modify:=False;
      end;
   end;
PROCEDURE ZOOM;
   var x1,y1,x2,y2,a,b,Cont:integer;
   begin
```

```
{   if True then exit;{}
   MouseCursor(Off);
   Grava_Cola(10); Cont:=0;
   x1:=GetMouseX; a:=x1;
   y1:=GetMouseY; b:=y1;
   repeat SetColor(15);
          Rectangle(x1,y1,a,b);
          repeat x2:=GetMouseX;
                 y2:=GetMouseY;
                 if x2<90  then x2:=90 ;
                 if x2>630 then x2:=630;
                 if y2<10  then y2:=10 ;
                 if y2>470 then y2:=470;
                 until (x2<>a)or(y2<>b);
          SetColor(0);
          Rectangle(x1,y1,a,b);
          a:=x2; b:=y2;
          Cont:=(Cont+1)mod(500);
          if Cont=0 then
             begin
             Grava_Cola(01);
             Grava_Cola(10);
             end;
          until button<>1;
    Grava_Cola(01);
    MouseCursor(On);
    end;
 PROCEDURE VERLABELS;
    Var A:       integer  ;
    begin
    MouseCursor(Off);
    Grava_Cola(10);
    MWindow(218,78,507,302);
    MouseCursor(On);
    A:=0; i:=0;
    Car:=#13;
    repeat if (Car=#13)or(button=1) then
              begin
              if i>=N then A:=0;
              MouseCursor(Off);
              Barra(215,75,7,'',295,230);
              Rectangle(218,78,507,302);
              OutTextXY(380,280,'<ENTER>-NEXT 10');
              for i:=10*A+1 to 10*(A+1) do
                  begin
                  if i<=N then
                     begin
                     k:=(i-1)mod 10;
```

```
                    str(i,aux);
                    OutTextXY(231,100+15*k,aux);
                    OutTextXY(240,100+15*k,' -->');
                    OutTextXY(275,100+15*k,Labels[i]);
                    end;
                end;
            A:=A+1; Car:=' ';
            MouseCursor(On) ;
            end;
        if keypressed then Car:=readkey;
        until (Car=#27)or(button=2);
   MouseCursor(Off);
   Grava_Cola(01);
   MouseCursor(On);
   MWindow(0,0,640,480);
   end;
PROCEDURE INTERLIGA(A:Vetor;NoF,NoD,ApRisc:integer);
   var a1:Vetor;          {se ApRisc:01-Esta' riscando}
       auxiliar:string[5];{          10-Esta' apagando}
   begin
   MouseCursor(Off);
   if ApRisc=01 then SetColor(CT) else SetColor(9);
   if (A[1]=0)and(ApRisc=01) then
       begin
       SetLineStyle(0,0,1);
       Grava_Cola(10);
       Barra(385,85,7,'',140,90);
       SetColor(1); MWindow(388,88,522,172); MouseCursor(On);
       Rectangle(388,88,522,172);
       str(NoF,aux); Car:=' ';
       str(NoD,auxiliar);
       OutTextXY(400,120,'Nao h'+aa+' Caminho');
       OutTextXY(405,135,'de '+aux+' Para '+auxiliar);
       Repeat if keypressed then Car:=readkey
              until (Car=#27)or(button=2);
       MWindow(0,0,640,480); MouseCursor(Off);
       Grava_Cola(01);
       end;
   if (A[1]<>0) then
       begin
       Liga(NoF,A[1]);
       i:=1;
       while ( A[i]<>NoD ) do
             begin
             if (BAtraso)and(ApRisc=01) then delay(1000);
             Liga(A[i],A[i+1]);
             i:=i+1;
             end
```

```
        end;
    MouseCursor(On);
    end;

END.
```