

```

1  {
2  Autor : Slawek Kolasinski
3  Data  : 04.2009
4  e-mail: skola@mimuw.edu.pl
5  }
6  Program Graph;
7  const
8    N = 10;
9  type
10   typ      = integer;
11   lista    = ^element;
12   wezel    = record
13     dane      : typ;
14     sasiedzi : lista;
15   end;
16   element = record
17     glowa : integer;
18     ogon  : lista;
19   end;
20   graf    = array[1..N] of wezel;

22  (***** OBSLUGA OPERACJI NA LISTACH *****)
23  procedure sl_inicjuj(var l : lista);
24  begin
25    l := nil;
26  end; { sl_inicjuj }

28  function sl_pusta(l : lista) : boolean;
29  begin
30    sl_pusta := (l = nil);
31  end; { sl_pusta }

33  function sl_glowa(l : lista) : integer;
34  begin
35    if not sl_pusta(l) then
36      sl_glowa := l^.glowa
37    else begin
38      writeln('glowa: pusta lista!');
39      halt;
40    end;
41  end; { sl_glowa }

43  function sl_ogon(l : lista) : lista;
44  begin
45    if not sl_pusta(l) then
46      sl_ogon := l^.ogon
47    else begin
48      writeln('ogon: pusta lista!');
49      halt;
50    end;
51  end; { sl_ogon }

53  function sl_usun(var l : lista) : integer;
54  var
55    czubek : lista;
56  begin
57    sl_usun := sl_glowa(l);
58    czubek := l;
59    l := sl_ogon(l);
60    dispose(czubek);

```

```

61  end; { sl_usun }

63  procedure sl_kasuj(var l : lista);
64  var
65    ignoruj : integer;
66  begin
67    while not sl_pusta(l) do ignoruj := sl_usun(l);
68  end; { sl_kasuj }

70  procedure sl_drukuj(l : lista);
71  begin
72    if not sl_pusta(l) then begin
73      write(sl_glowa(l));
74      l := sl_ogon(l);
75    end;
76    while not sl_pusta(l) do begin
77      write(' ', sl_glowa(l));
78      l := sl_ogon(l);
79    end;
80  end; { sl_drukuj }

82  procedure sl_dodaj(var l : lista; x : integer);
83  var
84    elt : lista;
85  begin
86    new(elt);
87    elt^.glowa := x;
88    elt^.ogon := l;
89    l := elt;
90  end; { sl_dodaj }

92  function sl_znajdz(l : lista; x : typ) : boolean;
93  begin
94    while (not sl_pusta(l)) and (sl_glowa(
95      l) <> x) do l := sl_ogon(l);
96  end; { sl_znajdz }

98  function sl_dlugosc(l : lista) : integer;
99  var
100   licznik : integer;
101  begin
102   licznik := 0;
103   while not sl_pusta(l) do begin
104     licznik := licznik + 1;
105     l := sl_ogon(l);
106   end;
107   sl_dlugosc := licznik;
108  end; { sl_dlugosc }

110  (***** OBSLUGA OPERACJI NA GRAFACH *****)
111  procedure gr_inicjuj(var g : graf);
112  var
113    i : integer;
114  begin
115    { Zainicjuj listy sasiedztwa na puste listy.
116      Danych nie ruszaj - moga zostac smieci }
117    for i := 1 to N do sl_inicjuj(g[i].sasiedzi);
118  end; { gr_inicjuj }

```

```

120 procedure gr_dodaj_krawedz(var g : graf; u,v : integer);
121 begin
122     sl_dodaj(g[u].sasiedzi, v);
123 end; { gr_dodaj_krawedz }

125 { Dodaje krawedz w obie strony }
126 procedure gr_dodaj_krawedz2(var g : graf; u,v : integer);
127 begin
128     gr_dodaj_krawedz(g,u,v);
129     if u <> v then gr_dodaj_krawedz(g,v,u);
130 end; { gr_dodaj_krawedz2 }

132 function gr_jest_krawedz(var g : graf; u,v : typ) : boolean;
133 begin
134     gr_jest_krawedz := sl_znajdz(g[u].sasiedzi, v);
135 end; { gr_jest_krawedz }

137 function gr_stopien(var g : graf; u : integer) : integer;
138 begin
139     gr_stopien := sl_dlugosc(g[u].sasiedzi);
140 end; { gr_stopien }

142 procedure gr_drukuj(var g : graf);
143 var
144     u : integer;
145 begin
146     for u := 1 to N do begin
147         write(u, '(' ,gr_stopien(g,u), '): ');
148         sl_drukuj(g[u].sasiedzi);
149         writeln;
150     end;
151 end; { gr_drukuj }

153 procedure gr_kasuj(var g : graf);
154 var
155     u : integer;
156 begin
157     for u := 1 to N do begin
158         sl_kasuj(g[u].sasiedzi);
159     end;
160 end; { gr_kasuj }

162 procedure gr_z_pliku(nazwa : string; var g : graf);
163 var
164     plik : text;
165     u,v : integer;
166 begin
167     assign(plik,nazwa);
168     reset(plik);
169     gr_inicjuj(g);
170     while not eof(plik) do begin
171         readln(plik, u, v);
172         if (1 <= u) and (1 <= v) and (u <= N) and (v <= N) then
173             gr_dodaj_krawedz(g,u,v)
174         else begin
175             writeln(
176                 'gr_z_pliku: Indeks wierzcholka poza zakresem!');
177             halt;
178         end;
179     end;
180 end;

```

```

179     close(plik);
180 end; { gr_z_pliku }

182 procedure gr_z_pliku2(nazwa : string; var g : graf);
183 var
184     plik : text;
185     u,v : integer;
186 begin
187     assign(plik,nazwa);
188     reset(plik);
189     gr_inicjuj(g);
190     while not eof(plik) do begin
191         readln(plik, u, v);
192         if (1 <= u) and (1 <= v) and (u <= N) and (
193             v <= N) then begin
194             if (not gr_jest_krawedz(g,u,v)) then gr_dodaj_krawedz2(
195                 g,u,v);
196             end else begin
197                 writeln(
198                     'gr_z_pliku: Indeks wierzcholka poza zakresem!');
199                 halt;
200             end;
201         close(plik);
202     end; { gr_z_pliku2 }

202 procedure gr_transponuj(var g,gt : graf);
203 var
204     u,v : integer;
205     l : lista;
206 begin
207     gr_inicjuj(gt);
208     for u := 1 to N do begin
209         l := g[u].sasiedzi;
210         while not sl_pusta(l) do begin
211             v := sl_glowa(l);
212             gr_dodaj_krawedz(gt,v,u);
213             l := sl_ogon(l);
214         end;
215     end;
216 end; { gr_transponuj }

218 var
219     g1,g2,gt : graf;
220 begin
221     gr_z_pliku('graf.txt',g1);
222     writeln('gr1');
223     gr_drukuj(g1);

225     gr_z_pliku2('graf.txt',g2);
226     writeln('gr2');
227     gr_drukuj(g2);

229     gr_transponuj(g1,gt);
230     writeln('gt');
231     gr_drukuj(gt);

233     gr_kasuj(g1);
234     gr_kasuj(g2);
235 end.

```