

```

1  {
2  Autor : Slawek Kolasinski
3  Data  : 02.2009
4  e-mail: skola@mimuw.edu.pl
5  }

7  Program FromFile;
8  type
9      typ      = integer;
10     lista    = ^element;
11     element  = record
12         glowa : typ;
13         ogon  : lista;
14     end;

16 { Inicjuje liste }
17 procedure inicjuj(var l : lista);
18 begin
19     l := nil;
20 end; { inicjuj }

22 { Sprawdza czy lista jest pusta }
23 function pusta(l : lista) : boolean;
24 begin
25     pusta := (l = nil);
26 end; { pusta }

28 { Zwraca wartosc zapisana w pierwszym elemencie listy
29   ale nie modyfikuje listy }
30 function glowa(l : lista) : typ;
31 begin
32     if not pusta(l) then
33         glowa := l^.glowa
34     else
35         begin
36             writeln('glowa: pusta lista!');
37             halt;
38         end;
39 end; { glowa }

41 { Zwraca ogon listy,
42   tzn. cala liste bez pierwszego elementu.
43   Nie modyfikuje listy. }
44 function ogon(l : lista) : lista;
45 begin
46     if not pusta(l) then
47         ogon := l^.ogon
48     else
49         begin
50             writeln('ogon: pusta lista!');

```

```

50     halt;
51     end;
52 end; { ogon }

54 { Usuwa pierwszy element z listy }
55 function usun(var l : lista) : typ;
56 var
57     czubek : lista;
58 begin
59     usun := glowa(l);
60     czubek := l;
61     l := ogon(l);
62     dispose(czubek);
63 end; { usun }

65 { Usuwa liste z pamieci. }
66 procedure kasuj(var l : lista);
67 var
68     ignoruj : typ;
69 begin
70     while not pusta(l) do ignoruj := usun(l);
71 end; { kasuj }

73 { Dodaje na poczatek listy nowy element }
74 procedure dodaj(var l : lista; x : typ);
75 var
76     elt : lista;
77 begin
78     new(elt);
79     elt^.glowa := x;
80     elt^.ogon := l;
81     l := elt;
82 end; { dodaj }

84 { Odwraca liste. }
85 function odwroc(l : lista) : lista;
86 var
87     wynik : lista;
88 begin
89     inicjuj(wynik);
90     while not pusta(l) do
91         begin
92             dodaj(wynik, glowa(l));
93             l := ogon(l);
94         end;
95     odwroc := wynik;
96 end; { odwroc }

98 { Scala dwie uporządkowane rosnaco listy.
99   W wyniku tworzy nowa liste. }

```

```

100 function scal(l1,l2 : lista) : lista;
101 var
102     wsk1,wsk2,wynik,rezsta : lista;
103     dane1,dane2           : typ;
104 begin
105     wsk1 := l1;
106     wsk2 := l2;
107     inicjuj(wynik);

109     while (not pusta(wsk1)) and (not pusta(wsk2)) do
110     begin
111         dane1 := glowa(wsk1);
112         dane2 := glowa(wsk2);
113         if dane1 < dane2 then
114         begin
115             dodaj(wynik, dane1);
116             wsk1 := ogon(wsk1);
117         end else begin
118             dodaj(wynik, dane2);
119             wsk2 := ogon(wsk2);
120         end
121     end;

123     if not pusta(wsk1) then
124         rezsta := wsk1
125     else
126         rezsta := wsk2;

128     while not pusta(reszta) do
129     begin
130         dodaj(wynik, glowa(reszta));
131         rezsta := ogon(reszta);
132     end;

134     scal := odwroc(wynik);
135     kasuj(wynik);
136 end; { scal }

138 { Drukuje na ekranie cala liste. }
139 procedure drukuj(l : lista);
140 var
141     wsk : lista;
142 begin
143     wsk := l;
144     if not pusta(wsk) then
145     begin
146         write(wsk^.glowa);
147         wsk := ogon(wsk);
148     end;
149     while not pusta(wsk) do

```

```

150     begin
151         write(' ', wsk^.glowa);
152         wsk := ogon(wsk);
153     end;
154 end; { drukuj }

156 { Wczytuje liczby z pliku do listy. }
157 function zpliku(sciezka : string) : lista;
158 var
159     plik : text;
160     wynik : lista;
161     liczba : typ;
162 begin
163     inicjuj(wynik);
164     assign(plik, sciezka);
165     reset(plik);
166     while not eof(plik) do
167     begin
168         readln(plik, liczba);
169         dodaj(wynik, liczba);
170     end;
171     close(plik);
172     zpliku := odwroc(wynik);
173     kasuj(wynik);
174 end;

176 var
177     l1,l2,l3 : lista;

179 begin
180     l1 := zpliku('a.txt');
181     l2 := zpliku('b.txt');

183     write('l1 = '); drukuj(l1); writeln(' ');
184     write('l2 = '); drukuj(l2); writeln(' ');

186     l3 := scal(l1,l2);
187     write('po scaleniu l3 = '); drukuj(l3);
                                                    writeln(' ');

189     kasuj(l1);
190     kasuj(l2);
191     kasuj(l3);
192 end.

```