

```

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

7  Program Stack;
8  type
9      typ      = integer;
10     stos     = ^element;
11     element  = record
12         wartosc   : typ;
13         poprzedni : stos;
14     end;
15 var
16     s1,s2 : stos;

18 procedure init(var s : stos);
19 begin
20     s := nil;
21 end; { init }

23 function pop(var s : stos) : typ;
24 var
25     czubek : stos;
26 begin
27     if s <> nil then
28     begin
29         pop := s^.wartosc;
30         czubek := s;
31         s := czubek^.poprzedni;
32         dispose(czubek);
33     end
34     else
35     begin
36         writeln('pop: pusty lista!');
37         halt;
38     end;
39 end; { pop }

41 procedure push(var s : stos; x : typ);
42 var
43     elt : stos;
44 begin
45     new(elt);
46     elt^.wartosc := x;
47     elt^.poprzedni := s;
48     s := elt;
49 end; { push }

```

```

51 function top(s : stos) : typ;
52 begin
53     if s <> nil then
54         top := s^.wartosc
55     else
56     begin
57         writeln('top: pusty lista!');
58         halt;
59     end;
60 end; { top }

62 function is_empty(s : stos) : boolean;
63 begin
64     is_empty := (s = nil);
65 end; { is_empty }

67 procedure print(s : stos);
68 var
69     ptr : stos;
70 begin
71     ptr := s;
72     if ptr <> nil then
73     begin
74         write(ptr^.wartosc);
75         ptr := ptr^.poprzedni;
76     end;
77     while ptr <> nil do
78     begin
79         write(' ', ptr^.wartosc);
80         ptr := ptr^.poprzedni;
81     end;
82 end;

84 begin
85     init(s1);
86     init(s2);

88     writeln('s1 pusty? ', is_empty(s1));
89     writeln('s2 pusty? ', is_empty(s2));

91     push(s1, 0);
92     push(s2, 9);
93     push(s1, 1);
94     push(s2, 8);
95     push(s1, 2);
96     push(s2, 7);
97     push(s1, 3);
98     push(s2, 6);
99     push(s1, 4);
100    push(s2, 5);

```

```

102 write('s1 = |'); print(s1); writeln('|');
103 write('s2 = |'); print(s2); writeln('|');

105 writeln('s1 pusty? ', is_empty(s1));
106 writeln('s2 pusty? ', is_empty(s2));

108 writeln('top(s1) = ', top(s1));
109 writeln('top(s2) = ', top(s2));

111 writeln('Zdejmuje ', pop(s1), ' z s1');
112 writeln('Zdejmuje ', pop(s1), ' z s1');
113 writeln('Zdejmuje ', pop(s1), ' z s1');
114 writeln('Zdejmuje ', pop(s2), ' z s2');

116 write('s1 = |'); print(s1); writeln('|');
117 write('s2 = |'); print(s2); writeln('|');

119 writeln('top(s1) = ', top(s1));
120 writeln('top(s2) = ', top(s2));

122 push(s1, 5);
123 push(s2, 4);
124 push(s1, 6);
125 push(s2, 3);
126 push(s1, 7);
127 push(s2, 2);
128 push(s1, 8);
129 push(s2, 1);
130 push(s1, 9);
131 push(s2, 0);

133 write('s1 = |'); print(s1); writeln('|');
134 write('s2 = |'); print(s2); writeln('|');

136 writeln('top(s1) = ', top(s1));
137 writeln('top(s2) = ', top(s2));

139 writeln('Zdejmuje ', pop(s1), ' z s1');
140 writeln('Zdejmuje ', pop(s1), ' z s1');
141 writeln('Zdejmuje ', pop(s1), ' z s1');
142 writeln('Zdejmuje ', pop(s1), ' z s1');
143 writeln('Zdejmuje ', pop(s1), ' z s1');
144 writeln('Zdejmuje ', pop(s1), ' z s1');
145 writeln('Zdejmuje ', pop(s1), ' z s1');

147 write('s1 = |'); print(s1); writeln('|');
148 write('s2 = |'); print(s2); writeln('|');

150 writeln('s1 pusty? ', is_empty(s1));

```

```

151 writeln('s2 pusty? ', is_empty(s2));
152 end.

154 {
155 Wynik dzia- ania programu na ekranie:
156
158 s1 pusty? True
159 s2 pusty? True
160 s1 = |4 3 2 1 0|
161 s2 = |5 6 7 8 9|
162 s1 pusty? False
163 s2 pusty? False
164 top(s1) = 4
165 top(s2) = 5
166 Zdejmuje 4 z s1
167 Zdejmuje 3 z s1
168 Zdejmuje 2 z s1
169 Zdejmuje 5 z s2
170 s1 = |1 0|
171 s2 = |6 7 8 9|
172 top(s1) = 1
173 top(s2) = 6
174 s1 = |9 8 7 6 5 1 0|
175 s2 = |0 1 2 3 4 6 7 8 9|
176 top(s1) = 9
177 top(s2) = 0
178 Zdejmuje 9 z s1
179 Zdejmuje 8 z s1
180 Zdejmuje 7 z s1
181 Zdejmuje 6 z s1
182 Zdejmuje 5 z s1
183 Zdejmuje 1 z s1
184 Zdejmuje 0 z s1
185 s1 = |
186 s2 = |0 1 2 3 4 6 7 8 9|
187 s1 pusty? True
188 s2 pusty? False

190 }

```