1. **type**
2. TCompareFunc = **function**(a, b: Integer): Boolean;
3. **procedure** Swap(**var** a, b: Integer);
4. **var**
5. temp: Integer;
6. **begin**
7. temp := a;
8. a := b;
9. b := temp;
10. **end**;
11. **function** DefaultCompare(a, b: Integer): Boolean;
12. **begin**
13. Result := a > b;
14. **end**;
15. **function** ReverseCompare(a, b: Integer): Boolean;
16. **begin**
17. Result := a < b;
18. **end**;
19. **procedure** InsertionSort(**var** arr: **array of** Integer; n: Integer; compare: TCompareFunc);
20. **var**
21. i, j, key: Integer;
22. **begin**
23. **for** i := 1 **to** n - 1 **do**
24. **begin**
25. key := arr[i];
26. j := i - 1;
27. **while** (j >= 0) **and** compare(arr[j], key) **do**
28. **begin**
29. arr[j + 1] := arr[j];
30. Dec(j);
31. **end**;
32. arr[j + 1] := key;
33. **end**;
34. **end**;
35. **procedure** Heapify(**var** arr: **array of** Integer; n, i: Integer; compare: TCompareFunc);
36. **var**
37. largest, left, right: Integer;
38. **begin**
39. largest := i;
40. left := 2 \* i + 1;
41. right := 2 \* i + 2;
42. **if** (left < n) **and** compare(arr[left], arr[largest]) **then**
43. largest := left;
44. **if** (right < n) **and** compare(arr[right], arr[largest]) **then**
45. largest := right;
46. **if** largest <> i **then**
47. **begin**
48. Swap(arr[i], arr[largest]);
49. Heapify(arr, n, largest, compare);
50. **end**;
51. **end**;
52. **procedure** HeapSort(**var** arr: **array of** Integer; n: Integer; compare: TCompareFunc);
53. **var**
54. i: Integer;
55. **begin**
56. **for** i := n **div** 2 - 1 **downto** 0 **do**
57. Heapify(arr, n, i, compare);
58. **for** i := n - 1 **downto** 1 **do**
59. **begin**
60. Swap(arr[0], arr[i]);
61. Heapify(arr, i, 0, compare);
62. **end**;
63. **end**;
64. **procedure** PrintArray(arr: **array of** Integer; n: Integer);
65. **var**
66. i: Integer;
67. **begin**
68. **for** i := 0 **to** n - 1 **do**
69. Write(arr[i], ' ');
70. Writeln;
71. **end**;
72. **procedure** ReadArrayFromFile(**var** arr: **array of** Integer; **const** fileName: string; **var** n: Integer);
73. **var**
74. f: TextFile;
75. temp: Integer;
76. **begin**
77. AssignFile(f, fileName);
78. Reset(f);
80. n := 0;
81. SetLength(arr, 0);
83. **while not** Eof(f) **do**
84. **begin**
85. Read(f, temp);
86. SetLength(arr, n + 1);
87. arr[n] := temp;
88. Inc(n);
89. **end**;
90. CloseFile(f);
91. **end**;
92. **var**
93. arr: **array of** Integer;
94. n, i: Integer;
95. fileName: string;
96. **begin**
97. Write('Введите имя файла: ');
98. ReadLn(fileName);
100. ReadArrayFromFile(arr, fileName, n);
101. **if** n = 0 **then**
102. **begin**
103. Writeln('Массив пуст. Проверьте файл.');
104. **Exit**;
105. **end**;
106. Writeln('Массив до сортировки вставками:');
107. PrintArray(arr, n);
108. InsertionSort(arr, n, DefaultCompare);
109. Writeln('Массив после сортировки вставками:');
110. PrintArray(arr, n);
111. Writeln('Массив до пирамидальной сортировки:');
112. PrintArray(arr, n);
113. HeapSort(arr, n, DefaultCompare);
114. Writeln('Массив после пирамидальной сортировки:');
115. PrintArray(arr, n);
116. **end**.