1. **program** Stones;
2.
3. **const**
4. MAX\_N = 30;
5. **type**
6. TIndex = 0..MAX\_N-1;
7. TStones = **array** [TIndex] **of** **Integer**;
8.
9. **var**
10. N: **Integer**;
11. P: TStones;
12. i: TIndex;
13. W, T, tOld: **LongInt**;
14. V: **LongInt**;
15.
16.
17. **procedure** PrintSolution(**const** s: **String**);
18. **var**
19. k: TIndex;
20. **begin**
21. Write(s);
22. WriteLn;
23. *// текущий вес*
24. Write(T:4,': ');
25. **for** k := 0 **to** N-1 **do**
26. **if** V **shr** k and 1 > 0 **then**
27. Write(P[k]:3, ' ');
28. **end**;
29.
30. **begin**
31. *// загрузка данных и вывод задачи на консоль*
32. ReadLn(W);
33. ReadLn(N);
34. **if** N > MAX\_N **then** Exit;
35. Write('Набрать сумму ', W:4, ' из чисел ');
36. **for** i := 0 **to** N-1 **do** **begin**
37. Read(P[i]);
38. Write(P[i]:3, ' ');
39. **end**;
40. WriteLn;
41.
42. tOld:=0;
43. **for** V := 0 **to** (1 **shl** N)-1 **do** **begin**
44. *// определение веса набора*
45. T := 0;
46. **for** i := 0 **to** N-1 **do**
47. **if** V **shr** i and 1 > 0 **then**
48. Inc(T, P[i]);
49.
50. *// все решения - было для отладки*
51. *//PrintSolution('');*
52.
53. *// вывод только нужных*
54. **if** T = W **then** **begin**
55. **if** T > tOld **then** **begin**
56. *// нашли первое решение*
57. PrintSolution('- мусор'#13'Решения:');
58. tOld := T;
59. **end**
60. **else**
61. PrintSolution('');
62. **end**
63. **else** **if** T < W **then** **begin**
64. *// может быть решением, если набрать не удастся*
65. **if** T > tOld **then** **begin**
66. PrintSolution('- мусор');
67. tOld := T;
68. **end**
69. **else** **if** T = tOld **then**
70. PrintSolution('');
71. *// случай T < tOld не выводим*
72. **end**;
73. **end**;
74. **end**.