- Model Solution

```
(* Solution to part (a) -- number of ways in which coins a, b, c, d can be used to create amount amt.
k is the parameter that controls which of a, b, c, d are available - required since they
cannot use lists.*)

let rec ways_helper amt a b c d k =
        if amt = 0 then 1
        else if amt < 0 then 0
        else match k with
        | 1 -> (coins (amt-a) a b c d k) + (coins amt a b c d (k+1))
        | 2 -> (coins (amt-b) a b c d k) + (coins amt a b c d (k+1))
        | 3 -> (coins (amt-c) a b c d k) + (coins amt a b c d (k+1))
        | 4 -> (coins (amt-d) a b c d k)
        | _ -> 0

let ways amt a b c d = ways_helper amt a b c d 1

(* Solution to part (b) -- best way to combine coins a, b, c, d to create amount amt.
'best' is defined by a weight function which is a parameter to  the main cost function*)

let weight x = 2*x;;
let weightI x = 100 - x;;

let rec cost_helper amt a b c d k f=
        if amt = 0 then 0
        else if amt < 0 then 0
        else match k with
        | 1 -> min ((f a) + (cost (amt-a) a b c d k f)) (cost amt a b c d (k+1) f)
        | 2 -> min ((f b) + (cost (amt-b) a b c d k f)) (cost amt a b c d (k+1) f)
        | 3 -> min ((f c) + (cost (amt-c) a b c d k f)) (cost amt a b c d (k+1) f)
```

```
    | 4 -> (f d) + (cost (amt-d) a b c d k f)
    | _ -> 0

let cost amt a b c d f = cost_helper a b c d 1 f
```

- Test Cases

1. 25 1 -5 4 -5 1
2. 39 10 -4 0 5 0
3. 40 17 10 -1 14 0
4. -2 -5 -5 -4 -1 1
5. 24 -3 12 0 4 0
6. 43 17 -1 14 -2 1
7. 49 11 22 0 4 0
8. 12 -4 3 -2 -1 0
9. 30 7 7 -3 12 1
10. 49 -5 16 15 17 0
11. -4 -5 -4 -5 -3 0
12. 33 9 3 7 7 0
13. 7 3 -1 -4 -1 0
14. 28 -1 9 14 -1 1
15. 42 6 19 9 -1 1
16. 14 1 -2 3 4 0
17. 18 2 -2 6 -2 1
18. -2 -5 -4 -2 -2 0
19. -5 -4 -5 -4 -3 0
20. 9 -2 -3 -2 2 0
21. 12 4 1 3 5 1
22. 34 9 6 15 2 0
23. 36 5 8 3 12 0
24. 36 6 18 16 13 1
25. 39 5 7 13 11 1
26. 50 8 3 4 19 0
27. 12 4 3 1 2 1
28. 46 1 16 20 21 1
29. 38 5 13 12 8 0
30. 44 10 2 7 16 1
31. 24 12 11 10 9 0
32. 45 1 8 11 16 0
33. 45 8 19 18 15 0

34. 49 6 5 20 15 0
35. 46 6 14 1 5 1
36. 50 10 17 5 23 1
37. 48 23 22 7 1 1
38. 36 2 6 9 1 0
39. 30 7 10 5 6 1
40. 40 6 18 3 12 1
41. 28 9 14 7 10 1
42. 43 21 5 9 2 1
43. 48 4 2 16 7 0
44. 28 5 1 6 14 0
45. 39 19 5 7 14 0
46. 43 10 16 7 18 0
47. 42 3 15 13 9 1
48. 25 2 7 12 5 0
49. 25 8 5 1 12 0
50. 36 14 17 10 4 0
51. 25 9 4 11 5 0
52. 35 14 4 7 16 1
53. 35 17 1 4 15 1
54. 44 6 15 17 3 1
55. 34 12 5 16 1 1
56. 50 10 23 20 22 0
57. 39 15 16 7 13 1
58. 42 9 5 21 18 1
59. 47 5 17 19 15 1
60. 29 6 8 5 9 0
61. 29 11 2 5 13 0
62. 32 5 1 6 16 1
63. 43 11 10 14 5 1
64. 48 19 13 2 23 0
65. 29 12 1 14 10 1
66. 27 8 10 4 2 0
67. 31 5 2 13 8 1
68. 49 22 12 20 2 1
69. 38 11 17 16 5 1
70. 19 8 4 9 1 0
71. 25 8 10 1 5 0
72. 25 6 9 5 2 1
73. 43 14 10 7 1 1
74. 47 10 17 20 4 0
75. 41 2 6 19 3 0
76. 49 3 17 10 20 0
77. 29 5 12 9 3 0

78. 26 2 12 5 11 1
79. 21 6 2 10 8 0
80. 45 11 12 13 3 1
81. 34 5 7 13 17 1
82. 49 8 22 24 13 0
83. 24 5 3 10 7 0
84. 43 1 10 18 3 1
85. 25 10 12 8 11 1
86. 31 8 15 1 5 0
87. 23 10 5 9 7 1
88. 37 2 11 16 8 0
89. 39 7 12 9 13 0
90. 43 1 21 18 11 0
91. 43 8 6 19 20 0
92. 33 8 10 4 2 1
93. 39 7 5 4 18 1
94. 43 21 4 20 17 1
95. 49 11 19 16 13 0
96. 49 7 12 5 11 1
97. 44 14 7 9 16 0
98. 12 5 4 6 2 0
99. 36 4 6 3 13 0
100. 39 12 6 4 7 1