# OUTPUT FOR SUM(50)

```
Enter the sum  you want : 50
Gen. 0 (0.00%): Max/Min/Avg Fitness(Raw) [0.00(0.00)/0.00(0.00)/0.00(0.00)]
Gen. 10 (5.00%): Max/Min/Avg Fitness(Raw) [0.00(0.00)/0.00(0.00)/0.00(0.00)]
Gen. 20 (10.00%): Max/Min/Avg Fitness(Raw) [0.01(0.01)/0.00(0.00)/0.00(0.00)]
Gen. 30 (15.00%): Max/Min/Avg Fitness(Raw) [0.02(0.02)/0.02(0.02)/0.02(0.02)]
Gen. 40 (20.00%): Max/Min/Avg Fitness(Raw) [2.00(2.00)/2.00(2.00)/2.00(2.00)]
Gen. 50 (25.00%): Max/Min/Avg Fitness(Raw) [1.51(2.00)/0.85(0.06)/1.26(1.26)]
Gen. 60 (30.00%): Max/Min/Avg Fitness(Raw) [1.34(2.00)/0.85(0.06)/1.11(1.11)]
Gen. 70 (35.00%): Max/Min/Avg Fitness(Raw) [1.76(2.00)/0.69(0.06)/1.47(1.47)]
Gen. 80 (40.00%): Max/Min/Avg Fitness(Raw) [1.70(2.00)/0.75(0.04)/1.42(1.42)]
Gen. 90 (45.00%): Max/Min/Avg Fitness(Raw) [1.55(2.00)/0.83(0.03)/1.29(1.29)]
Gen. 100 (50.00%): Max/Min/Avg Fitness(Raw) [1.68(2.00)/0.77(0.05)/1.40(1.40)]
Gen. 110 (55.00%): Max/Min/Avg Fitness(Raw) [1.34(2.00)/0.85(0.06)/1.12(1.12)]
Gen. 120 (60.00%): Max/Min/Avg Fitness(Raw) [1.77(2.00)/0.68(0.06)/1.48(1.48)]
Gen. 130 (65.00%): Max/Min/Avg Fitness(Raw) [1.29(2.00)/0.84(0.08)/1.08(1.08)]
Gen. 140 (70.00%): Max/Min/Avg Fitness(Raw) [1.30(2.00)/0.84(0.06)/1.08(1.08)]
Gen. 150 (75.00%): Max/Min/Avg Fitness(Raw) [1.51(2.00)/0.84(0.04)/1.26(1.26)]
Gen. 160 (80.00%): Max/Min/Avg Fitness(Raw) [1.46(2.00)/0.86(0.06)/1.22(1.22)]
Gen. 170 (85.00%): Max/Min/Avg Fitness(Raw) [1.68(2.00)/0.78(0.06)/1.40(1.40)]
Gen. 180 (90.00%): Max/Min/Avg Fitness(Raw) [1.63(2.00)/0.81(0.07)/1.36(1.36)]
Gen. 190 (95.00%): Max/Min/Avg Fitness(Raw) [1.42(2.00)/0.85(0.04)/1.19(1.19)]
Gen. 200 (100.00%): Max/Min/Avg Fitness(Raw) [1.42(2.00)/0.86(0.06)/1.18(1.18)]
Total time elapsed: 0.722 seconds.
- GenomeBase
      Score:                  2.000000
      Fitness:           1.419118

      Params:            {}

      Slot [Evaluator] (Count: 1)
            Name: eval_func - Weight: 0.50
      Slot [Initializator] (Count: 1)
            Name: G1DListInitializatorInteger - Weight: 0.50
            Doc:  Integer initialization function of G1DList

   This initializator accepts the *rangemin* and *rangemax* genome parameters.


      Slot [Mutator] (Count: 1)
            Name: G1DListMutatorSwap - Weight: 0.50
            Doc:  The mutator of G1DList, Swap Mutator

   .. note:: this mutator is :term:`Data Type Independent`


      Slot [Crossover] (Count: 1)
            Name: G1DListCrossoverSinglePoint - Weight: 0.50
            Doc:  The crossover of G1DList, Single Point

   .. warning:: You can't use this crossover method for lists with just one element.



- G1DList
      List size:   20
      List:        [1, 1, 16, 1, 5, 1, 0, 1, 0, 1, 13, 0, 0, 1, 1, 1, 5, 1, 1, 0]


50
```

# OUTPUT FOR SUM(100)

```
Enter the sum  you want: 100
Gen. 0 (0.00%): Max/Min/Avg Fitness(Raw) [0.00(0.00)/0.00(0.00)/0.00(0.00)]
Gen. 10 (5.00%): Max/Min/Avg Fitness(Raw) [0.00(0.00)/0.00(0.00)/0.00(0.00)]
Gen. 20 (10.00%): Max/Min/Avg Fitness(Raw) [0.02(0.02)/0.01(0.01)/0.02(0.02)]
Gen. 30 (15.00%): Max/Min/Avg Fitness(Raw) [2.23(2.00)/0.00(0.08)/2.06(1.86)]
Gen. 40 (20.00%): Max/Min/Avg Fitness(Raw) [1.94(2.00)/0.27(0.02)/1.62(1.62)]
Gen. 50 (25.00%): Max/Min/Avg Fitness(Raw) [1.54(2.00)/0.83(0.02)/1.28(1.28)]
Gen. 60 (30.00%): Max/Min/Avg Fitness(Raw) [1.83(2.00)/0.56(0.02)/1.52(1.52)]
Gen. 70 (35.00%): Max/Min/Avg Fitness(Raw) [1.65(2.00)/0.78(0.02)/1.37(1.37)]
Gen. 80 (40.00%): Max/Min/Avg Fitness(Raw) [1.59(2.00)/0.81(0.02)/1.33(1.33)]
Gen. 90 (45.00%): Max/Min/Avg Fitness(Raw) [2.05(2.00)/0.00(0.02)/1.75(1.71)]
Gen. 100 (50.00%): Max/Min/Avg Fitness(Raw) [1.88(2.00)/0.44(0.02)/1.57(1.57)]
Gen. 110 (55.00%): Max/Min/Avg Fitness(Raw) [2.23(2.00)/0.00(0.11)/2.06(1.86)]
Gen. 120 (60.00%): Max/Min/Avg Fitness(Raw) [1.48(2.00)/0.84(0.02)/1.23(1.23)]
Gen. 130 (65.00%): Max/Min/Avg Fitness(Raw) [1.67(2.00)/0.77(0.02)/1.39(1.39)]
Gen. 140 (70.00%): Max/Min/Avg Fitness(Raw) [1.84(2.00)/0.58(0.08)/1.53(1.53)]
Gen. 150 (75.00%): Max/Min/Avg Fitness(Raw) [1.60(2.00)/0.81(0.02)/1.33(1.33)]
Gen. 160 (80.00%): Max/Min/Avg Fitness(Raw) [1.60(2.00)/0.81(0.02)/1.33(1.33)]
Gen. 170 (85.00%): Max/Min/Avg Fitness(Raw) [1.59(2.00)/0.81(0.02)/1.33(1.33)]
Gen. 180 (90.00%): Max/Min/Avg Fitness(Raw) [1.07(2.00)/0.75(0.02)/0.89(0.89)]
Gen. 190 (95.00%): Max/Min/Avg Fitness(Raw) [1.65(2.00)/0.78(0.02)/1.38(1.38)]
Gen. 200 (100.00%): Max/Min/Avg Fitness(Raw) [1.37(2.00)/0.84(0.02)/1.14(1.14)]
Total time elapsed: 0.718 seconds.
- GenomeBase
      Score:                  2.000000
      Fitness:           1.373069

      Params:            {}

      Slot [Evaluator] (Count: 1)
            Name: eval_func - Weight: 0.50
      Slot [Initializator] (Count: 1)
            Name: G1DListInitializatorInteger - Weight: 0.50
            Doc:  Integer initialization function of G1DList

   This initializator accepts the *rangemin* and *rangemax* genome parameters.


      Slot [Mutator] (Count: 1)
            Name: G1DListMutatorSwap - Weight: 0.50
            Doc:  The mutator of G1DList, Swap Mutator

   .. note:: this mutator is :term:`Data Type Independent`


      Slot [Crossover] (Count: 1)
            Name: G1DListCrossoverSinglePoint - Weight: 0.50
            Doc:  The crossover of G1DList, Single Point

   .. warning:: You can't use this crossover method for lists with just one element.



- G1DList
      List size:   20
      List:        [1, 9, 46, 1, 1, 1, 1, 9, 9, 12, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]


100
```

# OUTPUT FOR SUM(150)

```
Enter the sum  you want 150
Gen. 0 (0.00%): Max/Min/Avg Fitness(Raw) [0.00(0.00)/0.00(0.00)/0.00(0.00)]
Gen. 10 (5.00%): Max/Min/Avg Fitness(Raw) [0.00(0.01)/0.00(0.00)/0.00(0.00)]
Gen. 20 (10.00%): Max/Min/Avg Fitness(Raw) [0.03(0.20)/0.02(0.01)/0.02(0.02)]
Gen. 30 (15.00%): Max/Min/Avg Fitness(Raw) [1.55(2.00)/0.83(0.03)/1.29(1.29)]
Gen. 40 (20.00%): Max/Min/Avg Fitness(Raw) [1.24(2.00)/0.82(0.02)/1.04(1.04)]
Gen. 50 (25.00%): Max/Min/Avg Fitness(Raw) [1.77(2.00)/0.67(0.04)/1.47(1.47)]
Gen. 60 (30.00%): Max/Min/Avg Fitness(Raw) [1.11(2.00)/0.77(0.02)/0.93(0.93)]
Gen. 70 (35.00%): Max/Min/Avg Fitness(Raw) [1.57(2.00)/0.83(0.04)/1.31(1.31)]
Gen. 80 (40.00%): Max/Min/Avg Fitness(Raw) [1.88(2.00)/0.46(0.02)/1.56(1.56)]
Gen. 90 (45.00%): Max/Min/Avg Fitness(Raw) [2.11(2.00)/0.00(0.04)/1.85(1.76)]
Gen. 100 (50.00%): Max/Min/Avg Fitness(Raw) [0.85(2.00)/0.63(0.02)/0.71(0.71)]
Gen. 110 (55.00%): Max/Min/Avg Fitness(Raw) [1.41(2.00)/0.85(0.04)/1.17(1.17)]
Gen. 120 (60.00%): Max/Min/Avg Fitness(Raw) [1.01(2.00)/0.72(0.03)/0.84(0.84)]
Gen. 130 (65.00%): Max/Min/Avg Fitness(Raw) [1.52(2.00)/0.84(0.05)/1.27(1.27)]
Gen. 140 (70.00%): Max/Min/Avg Fitness(Raw) [1.17(2.00)/0.80(0.03)/0.98(0.98)]
Gen. 150 (75.00%): Max/Min/Avg Fitness(Raw) [1.50(2.00)/0.84(0.03)/1.25(1.25)]
Gen. 160 (80.00%): Max/Min/Avg Fitness(Raw) [1.32(2.00)/0.84(0.04)/1.10(1.10)]
Gen. 170 (85.00%): Max/Min/Avg Fitness(Raw) [1.19(2.00)/0.80(0.02)/0.99(0.99)]
Gen. 180 (90.00%): Max/Min/Avg Fitness(Raw) [1.34(2.00)/0.84(0.03)/1.11(1.11)]
Gen. 190 (95.00%): Max/Min/Avg Fitness(Raw) [1.46(2.00)/0.85(0.03)/1.21(1.21)]
Gen. 200 (100.00%): Max/Min/Avg Fitness(Raw) [1.47(2.00)/0.85(0.03)/1.23(1.23)]
Total time elapsed: 0.714 seconds.
- GenomeBase
      Score:                  2.000000
      Fitness:           1.472646

      Params:            {}

      Slot [Evaluator] (Count: 1)
            Name: eval_func - Weight: 0.50
      Slot [Initializator] (Count: 1)
            Name: G1DListInitializatorInteger - Weight: 0.50
            Doc:  Integer initialization function of G1DList

   This initializator accepts the *rangemin* and *rangemax* genome parameters.


      Slot [Mutator] (Count: 1)
            Name: G1DListMutatorSwap - Weight: 0.50
            Doc:  The mutator of G1DList, Swap Mutator

   .. note:: this mutator is :term:`Data Type Independent`


      Slot [Crossover] (Count: 1)
            Name: G1DListCrossoverSinglePoint - Weight: 0.50
            Doc:  The crossover of G1DList, Single Point

   .. warning:: You can't use this crossover method for lists with just one element.



- G1DList
      List size:   20
      List:        [5, 0, 4, 37, 0, 5, 5, 0, 5, 0, 4, 0, 0, 0, 19, 5, 5, 5, 26, 25]


150
```