Sommario

[Black Box 2](#_Toc138599893)

[Compute Calories 2](#_Toc138599894)

[Compute Fee 2](#_Toc138599895)

[Compute Water Timing 4](#_Toc138599896)

[Show Picture 5](#_Toc138599897)

[White Box 7](#_Toc138599898)

[While Loop 7](#_Toc138599899)

[For Loop 7](#_Toc138599900)

[Array 7](#_Toc138599901)

[Compute Tax (already done) 8](#_Toc138599902)

[String Copy 8](#_Toc138599903)

# Black Box

## Compute Calories

CRITERIA

weightProteins

weightFats

weightCarbohidrates

PREDICATES

weightProteins sign <0,>0

weightFats sign <0,>0

weightCarbohidrates sign <0,>0

BOUNDARIES

weightProteins

[minint,-1] [0,maxint]

weightFats

[minint,-1] [0,maxint]

weightCarbohidrates

[minint,-1] [0,maxint]

EQUIVALENCE CLASSES AND TESTING

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| weightProteins | weightFats | weightCarbohidrates | Valid | Test Cases |
| [minint,-1] | \* | \* | I | T1(-8,…,…)->Error  TB(-1,..,…)->Error |
| \* | [minint,-1] | \* | I | T2(…,-8,…)->Error  TB(…,-1,…)->Error |
| \* | \* | [minint,-1] | I | T3(…,…,-8)->Error  TB(…,..,-1)->Error |
| [0,maxint] | [0,maxint] | [0,maxint] | V | T4(0,0,0)->0  TB for the others |
| [0,maxint] | [0,maxint] | [0,maxint] | V | T5(4,2,5)->54 |
| [0,maxint] | [0,maxint] | [0,maxint] | V | T6(0,2,5)->38 |
| [0,maxint] | [0,maxint] | [0,maxint] | V | T7(4,0,5)->36 |
| [0,maxint] | [0,maxint] | [0,maxint] | V | T8(4,2,0)->34 |

## Compute Fee

CRITERIA

Duration

Minrate

Minrate2

PREDICATES

Duration sign

Minrate sign

Minrate2 sign

BOUNDARIES

Duration

[minint,-1] [0,30] [31,90] [91,maxint]

Minrate

[minint,-1] [0,maxint]

Minrate2

[minint,-1] [0,maxint]

EQUIVALENCE CLASSES AND TEST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Duration | Minrate | Minrate2 | Valid | Test Cases |
| [minint,-1] | \* | \* | I | T1(-8,…,…)->Error  TB(-1,…,…)->Error |
| \* | [minint,-1] | \* | I | T2(…,-8,…)->Error  TB(…,-1,…)->Error |
| \* | \* | [minint,-1] | I | T3(…,…,-8)->Error  TB(…,…,-1)->Error |
| [0,30] | [0,maxint] | [0,maxint] | V | T4(20,1,5)->0  TB(30,0,0)->0 |
| [31,90] | [0,maxint] | [0,maxint] | V | T5(35,10,20)->50  TB(90,10,20)->600 |
| [91,maxint] | [0,maxint] | [0,maxint] | V | T6(95,10,20)->700  TB(91,10,20)->620 |

## Compute Water Timing

CRITERIA

Start

Duration

Ntimes  
PREDICATES

Start sign <0,>0

Duration sign <0,>0

Ntimes 1,2,3, !=1,2,3

BOUNDARIES

Start

[minint,-1] [0,1440] [1441,maxint]

Duration

[minint,-1] [0,1140] [1441,maxint]

NTimes

[minint,0] [1] [2] [3] [4,maxint]  
EQUIVALENCE CLASSES AND TEST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start | Duration | Ntimes | Valid | Test Cases |
| [minint,-1] | \* | \* | I | T1(-8,…,…)->Error  TB(-1,…,…)->Error |
| [1441,maxint] | \* | \* | I | T2(1500,…,…)->Error  TB(1441,…,…)->Error |
| \* | [minint,-1] | \* | I | T3(…,-8,…)->Error  TB(…,-1,…)->Error |
| \* | [1441,maxint] | \* | I | T4(…,1500,…)->Error  TB(…,1441,…)->Error |
| \* | \* | [minint,0] | I | T5(…,…,-8)->Error  TB(…,…,0)->Error |
| \* | \* | [4,maxint] | I | T6(…,…,8)->Error  TB(…,…,4)->Error |
| [0,1440] | [0,1440] | [1] | V | T7(60,20,1)->[60,80,-1,-1,-1,-1]  TB(0,0,1)->[0,0,-1,-1,-1,-1]  TB(1440,1440,1)->Error |
| [0,1440] | [0,1440] | [2] | V | T8(60,20,2)->[60,80,780,800,-1,-1]  TB(0,0,2)->[0,0,0,0,-1,-1]  TB(1440,1440,2)->Error |
| [0,1440] | [0,1440] | [3] | V | T9(60,20,3)->[60,80,540,560,1020,1040]  TB(0,0,3)->[0,0,0,0,0,0]  TB(1440,1440,3)->Error |

## Show Picture

CRITERIA

Lat

Long

\*filename

PREDICATES

Lat -1,0,1,!=-1,0,1

Long -1,0,1,!=-1,0,1

\*filename in windows format, != windows format

BOUNDARIES

Lat

[minint,-2] [-1,1] [2,maxint]

Long

[minint,-2] [-1,1] [2,maxint]

\*filename

Windows format != windows format

EQUIVALENCE CLASSES AND TEST

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lat | Long | Is In Windows Format | Valid | Test Cases |
| [minint,-2] | \* | \* | V | T1(-8,…,…)->0  TB(-2,…,…)->0 |
| \* | [minint,-2] | \* | V | T2(…,-8,…)->0  TB(…,-2,…)->0 |
| [2,maxint] | \* | \* | V | T3(8,…,…)->0  TB(2,…,…)->0 |
| \* | [2,maxint] | \* | V | T4(…,8,…)->0  TB(…,2,…)->0 |
| \* | \* | F | I | T5(…,…,’dew’)->Error  TB(…,…,’C:\pictures\a.jpg’)->Error |
| [-1,1] | [-1,1] | T | V | T6(0,0, ’C:/pictures/a.jpg’)->1  TB(-1,-1, ’C:\pictures\a.jpg’)->1  TB(1,1, ’C:\pictures\a.jpg’)->1 |
| [-1,1] | [-1,1] | T | V | T7(0,1, ’C:\pictures\a.jpg’)->1  TB(1,-1, ’C:\pictures\a.jpg’)->1 |
| [-1,1] | [-1,1] | T | V | T8(1,0, ’C:\pictures\a.jpg’)->Error  TB(-1,1, ’C:\pictures\a.jpg’)->1 |

## Can Participate

CRITERIA

Teaching\_hours

Different\_courses

NA

EA

PREDICATES

Teaching\_hours <0,>0

Different\_courses <0,>0

NA <0,>0

NA\_normalized =NA,=NA\*10/EA

EA <0,>0

Formula1 🡪 teach:hours>1500 and different\_courses > 3 and NA\_normalized > 40

BOUNDARIES

Teaching\_Hours

[minint,-1] [0,1500] [1501,maxint]

Different\_Courses

[minint,-1] [0,3] [4,maxint]

NA

[minint,-1] [0,maxint]

NA\_Normalized

[0,40] [41,maxint]

EA

[minint,-1] [0,10] [11,maxint]

EQUIVALENCE CLASSES AND TEST

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Teaching\_Hours | Different\_Courses | NA | NA\_normalized | EA | Formula1 | Valid | Test Cases |
| [0,1500] | [0,3] | [0,maxint] | [0,40] | [0,10] | F | V | T1(1200,2,10,5)->false  TB(1500,3,0,10)->false |
| [0,1500] | [0,3] | [0,maxint] | [0,40] | [11,maxint] | F | V | T2(1200,2,10,15)->false  TB(1500,3,44,11)->false |
| [0,1500] | [0,3] | [0,maxint] | [41,maxint] | [0,10] | F | V | T3(1200,2,45,5)->false  TB(0,0,41,10)->false |
| [0,1500] | [0,3] | [0,maxint] | [41,maxint] | [11,maxint] | F | V | T4(1200,2,80,15)->false  TB(1500,3,46,11)->false |
| [0,1500] | [4,maxint] | [0,maxint] | [0,40] | [0,10] | F | V | T5(1200,5,30,5)->false  TB(1500,4,40,10)->false |
| [0,1500] | [4,maxint] | [0,maxint] | [0,40] | [11,maxint] | F | V | T6(1200,5,44,15)->false  TB(1500,4,44,11)->false |
| [0,1500] | [4,maxint] | [0,maxint] | [41,maxint] | [0,10] | F | V | T7(1200,5,50,5)->false  TB(1500,4,41,10)->false |
| [0,1500] | [4,maxint] | [0,maxint] | [41,maxint] | [11,maxint] | F | V | T8(1200,5,70,15)->false  TB(1500,4,44,11)->false |
| [1501,maxint] | [0,3] | [0,maxint] | [0,40] | [0,10] | F | V | T9(1600,2,30,5)->false  TB(1501,3,40,10)->false |
| [1501,maxint] | [0,3] | [0,maxint] | [0,40] | [11,maxint] | F | V | T10(1600,2,40,15)->false  TB(1501,3,44,11)->false |
| [1501,maxint] | [0,3] | [0,maxint] | [41,maxint] | [0,10] | F | V | T11(1600,2,50,5)->false  T12(1501,3,41,10)->false |
| [1501,maxint] | [0,3] | [0,maxint] | [41,maxint] | [11,maxint] | F | V | T12(1600,2,70,15)->false  TB(1501,3,44,11)->false |
| [1501,maxint] | [4,maxint] | [0,maxint] | [0,40] | [0,10] | F | V | T13(1600,5,30,5)->false  TB(1501,4,40,10)->false |
| [1501,maxint] | [4,maxint] | [0,maxint] | [0,40] | [11,maxint] | F | V | T14(1600,5,20,15)->false  TB(1501,4,44,11)->false |
| [1501,maxint] | [4,maxint] | [0,maxint] | [41,maxint] | [0,10] | T | V | T15(1600,5,50,5)->true  TB(1501,4,41,10)->true |
| [1501,maxint] | [4,maxint] | [0,maxint] | [41,maxint] | [11,maxint] | T | V | T16(1600,5,100,15)->true  TB(1501,4,45,11)->true |
| [minint,-1] | \* | \* | \* | \* | \* | I | T17(-8,…,…,…)->Error  TB(-1,…,…,…)->Error |
| \* | [minint,-1] | \* | \* | \* | \* | I | T18(…,-8,…,…)->Error  TB(…,-1,…,…)->Error |
| \* | \* | [minint,-1] | \* | \* | \* | I | T19(…,…,-8,…)->Error  TB(…,…,-1,…)->Error |
| \* | \* | \* | \* | [minint,-1] | \* | I | T20(…,…,…,-8)->Error  TB(…,…,…,-1)->Error |

## Luggage

CRITERIA

nLuggage length1, width1, depth1 length2, width2, depth2 weight1, weight2

PREDICATES

Sign of all

BOUNDARIES

nLuggage

[minint,-1] [0,2] [3,maxint]

Length,…

[minint,-1] [0,maxint]

Weight1 and 2

[minint,-1] [0,maxint]

Dim\_sum <=300

T F

Tot\_weight <=30

T F

EQUIVALENCE CLASSES AND TEST

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nluggage | Length1 | Width1 | Depth1 | Length2 | Width2 | Depth2 | Weight1 | Weight2 | Dimsum<=300 | Totweight<=30 | Valid | Test Cases |
| [3,maxint] | \* | \* | \* | \* | \* | \* | \* | \* | \* | \* | I | T1(5,…)->error  TB(3,…)->error |
| [minint,-1] | \* | \* | \* | \* | \* | \* | \* | \* | \* | \* | I | T2(-8,…)->error  TB(-1,…)->error |
| \* | [minint,-1] | \* | \* | \* | \* | \* | \* | \* | \* | \* | I | . |
| \* | \* | [minint,-1] | \* | \* | \* | \* | \* | \* | \* | \* | I | . |
| \* | \* | \* | [minint,-1] | \* | \* | \* | \* | \* | \* | \* | I | . |
| \* | \* | \* | \* | [minint,-1] | \* | \* | \* | \* | \* | \* | I | . |
| \* | \* | \* | \* | \* | [minint,-1] | \* | \* | \* | \* | \* | I | . |
| \* | \* | \* | \* | \* | \* | [minint,-1] | \* | \* | \* | \* | I | . |
| \* | \* | \* | \* | \* | \* | \* | [minint,-1] | \* | \* | \* | I | . |
| \* | \* | \* | \* | \* | \* | \* | \* | [minint,-1] | \* | \* | I | . |
| [0,2] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | T | T | V | T11(0,..)->true |
| [0,2] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] |  |  |  | T12(1,1,1,1,15,…,)->true  TB(1,100,100,100,30,..)->true |
| [0,2] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | T | T | V | T13(2,10,10,10,10,10,10,10,10)->true  TB(2,100,100,100,10,100,100,100,10)->true |
| [0,2] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | F | \* | V | T14(2,150,150,150,10,150,150,150,10)->false  TB(2,101,100,100,10,101,100,100,10)->false |
| [0,2] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | F | T | V | T15(2,150,100,100,10,150,100,100,15)->false  TB(2,101,100,100,10,101,100,100,10)->false |
| [0,2] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | [0,maxint] | \* | F | V | T16(2,10,10,10,20,10,10,10,20)->false  TB(2,10,10,10,15,10,10,10,16)->false |

# White Box

## While Loop

|  |  |  |  |
| --- | --- | --- | --- |
| Coverage | N of Test to obtain 100% coverage | Coverage Obtained with test case defined | Test Cases |
| Node | 2 | 100% | T1-T2 |
| Edge Coverage | 2 | 100% | T1-T2 |
| Multiple Condition Coverage | 4 | 100% | TT T1  FF T3  TF T2  FT T4 |
| Loop Coverage | 3 | 100% | T1 enter one  T2 try no enter  T5 enter many |
| Path Coverage | 2\*1\*2=4, so it is feasible | 100% | T1-T2-T4 |

T1(-1,-1,-6) T2(1,1,1) T3(2,1,3) T4(2,1,0) T5(-5,-5,-5)

## For Loop

|  |  |  |  |
| --- | --- | --- | --- |
| Coverage Type | N test to obtain 100% coverage | Coverage obtained with test cases defined | Test Cases |
| Node | 2 | 100% | T1-T2 |
| Edge | 3 | 100% | T1-T2-T3 |
| Multiple Condition line 7 | 4 | 100% | TT T1  TF T4  FT T2  FF T3 |
| Loop line 6 | 3 | 100% | Try no enter T5  Enter one T6  Enter many T2 |
| Loop line 9 | 3 | 100% | Try no enter T6  Enter one T7  Enter many T2 |
| Path | x\*x | Not feasible | - |

T1(-1,5) T2(5,5) T3(40,30) T4(30,40) T5(0,0) T6(1,0) T7(2,0)

## Array

|  |  |  |  |
| --- | --- | --- | --- |
| Coverage type | Number of test cases to obtain 100% coverage | Coverage obtained with test cases defined | Test Cases Defined |
| Node | 1 | 100% | T1 |
| Edge | 1 | 100% | T1 |
| Multiple Condition line 4 | 4 in theory but TF not feasible if i<array.lenth-1 will be also <MAXINT | 100% | TT T1 TF T5  FT T1  FF T4 |
| Loop line 4 | 3 | 100% | Try no enter T3  Enter one T2  Enter many T1 |
| Path | 2^(array.length-1) | Since length of array can be very high coverage is close to zero |  |

T1([3,2,1]) T2([1,2]) T3([1]) array of maxint length 🡺 T4([1,2,3,…,])

Array longer than maxint 🡺 T5([1,2,3,…])

## Compute Tax (already done)

## String Copy

|  |  |  |  |
| --- | --- | --- | --- |
| Coverage type | Number of test cases to obtain 100% coverage | Coverage obtained with test cases defined | Test Cases Defined |
| Node | 1 | 100% | T1 |
| Edge | 1 | 100% | T1 |
| Multiple Condition | No multiple condition | - | - |
| Loop line 5 | 3 in theory nut not feasible | 33% only enter many | Any test case |
| Loop line 7 | 3 in theory nut not feasible | 33% only enter many | Any test case |
| Loop line 8 | 3 in theory nut not feasible | 33% only enter many | Any test case |
| Path | 11\*2^(9\*9) | Not feasible | - |

T1(“ciao”,”c”,”ciao”,…)

## HeapSort

|  |  |  |  |
| --- | --- | --- | --- |
| Coverage Type | Number of test cases to obtain 100% coverage | Coverage obtained with test cases defined | Test Cases defined |
| Node | 1 | 100% | T1 |
| Edge | 1 | 100% | T1 |
| Multiple Condition Line 10 | 4 in theory, but covered by the loop | 100% | TT T1  TF T1  FT T2  FF T2 |
| Multiple Condition Line 12 | 4 in theory, but covered by the loop | 100% | TT T1  TF T1  FT T2  FF T2 |
| Loop Line 3 | 3 | 100% | Try no enter T3  Enter one T4  Enter many T1 |
| Loop Line 18 | 3 | 66%, since dowhile, try no enter not possible | Enter one T4  Enter many T1 |
| Path | 4^(?)^no, depend on the input | Since length if the array (no) can be very hugh, close to zero | - |

T1([1,2,3,4,5],5) T2([5,4,3,2,1],5) T3([],0) T4([1],1)

## Binary Search

|  |  |  |  |
| --- | --- | --- | --- |
| Coverage type | Number of test cases to obtain 100% coverage | Coverage obtained with test cases defined | Test Cases |
| Node | 3 | 100% | T1-T2-T3 |
| Edge | 3 | 100% | T1-T2-T3 |
| Multiple Condition | No mul. Cond. | \* | \* |
| Loop line 4 | 3 in theory but try no enter not possible | 100% | Try no Enter T5  Enter one T4  Enter many T1 |
| Path | (2\*2\*2)^? | Not feasible, since the number of paths depends on the number of serches | - |

T1([1,2,3,4,5,6,7,8,9,10,11],11,9) T2([1,2,3,4,5,6,7,8,9,10,11],11,2)

T3([1,2,3,4,5],5,40) T4([1,2,3],3,2) T5([],0,2)