**Problem7：奥步戰術(17%)**

在黑暗算法界中，使用奧步解題似乎已經漸漸成為主流。雖然使用奧步將漸漸使人走向魔路，最後被內心的虛無吞噬，不過這不是今天的問題。考慮在某個考試中，有*n* 道題目，而總答題時間為*T*。對於每題都只有三種可能：

1. 正解能得到全對的分數(得2 分)

2. 奧步能拿到半對(得1 分)

3. 放棄的話當然就沒分囉(0 分)

而對每題來說，要達到這三種分數所需花的時間皆不同，所有題目拿0 分都不用花費時間；在題目*i* 使用奧步拿半對所需時間為*Hi*，要寫正解所需時間為*Ci* ，其中對於任何題目*i*，必有滿足0<*Hi*<*Ci*。試問：在時間*T* 內，用最佳的答題方式，最多可以拿幾分？

**輸入說明：**

輸入檔第一行說明有幾組測試資料，第二行有兩個整數*n* 和*T*，分別代表有幾題，以及總作答時間。接下來*n* 行每行有兩個整數*Ci* 和*Hi*，代表第*i* 題寫正解需要時間*Ci*，寫奧步需要時間*Hi*。其中：

 題目總數n≤100000

 答題所需時間1≤*Hi*，*Ci*≤1000000

 總作答時間0<*T*≤1000000000

**輸出說明：**

每個測試範例請輸出一個整數，代表最大得分。

**輸入範例：**

2

5 12

4 3

6 2

5 3

4 3

5 2

4 10

5 3

6 5

3 1

4 3

**輸出範例：**

6

5

**程式碼：**

Imports System.IO

Public Class Form1

Dim fr As New FileInfo("test.txt")

Dim fw As New FileInfo("result.txt")

Dim sw As StreamWriter = fw.CreateText

Dim ia, ic(1), n, x, a2(2), oa, ob As Integer 'ia=幾組資料 ic=題數,時間 n and x(用來計數)

Dim ib As String 'ib(用來取每行資料方便分解)

Dim a1(2), min As Double

Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

If fr.Exists = False Then

End

ElseIf fw.Exists = False Then

fw.Create()

End If

Dim sr As StreamReader = fr.OpenText

ia = sr.ReadLine

For i = 1 To ia

ib = Trim(sr.ReadLine)

ReDim ic(1) : n = 0

For j = 1 To Len(ib) '取題數and時間

If Microsoft.VisualBasic.Mid(ib, j, 1) <> " " Then

ic(n) &= Microsoft.VisualBasic.Mid(ib, j, 1)

Else : n += 1

End If

Next

ReDim a1(ic(0) - 1), a2(ic(0) - 1)

Dim id(ic(0) - 1, 1) As Integer : n = 0

For j = 1 To ic(0) '取寫正解時間and寫奧步時間

ib = Trim(sr.ReadLine) : n = 0

For k = 1 To Len(ib)

If Microsoft.VisualBasic.Mid(ib, k, 1) <> " " Then

id(j - 1, n) &= Microsoft.VisualBasic.Mid(ib, k, 1)

Else : n += 1

End If

Next

Next

oa = 0 : ob = 0

min = id(0, 0) / 2

For j = 0 To ic(0) - 1

a1(j) = id(j, 0) / 2

If a1(j) < min Then min = a1(j)

Next

For j = 0 To ic(0) - 1

a2(j) = id(j, 1)

If a2(j) < min Then min = a2(j)

Next

2: For j = 0 To ic(0) - 1

If a1(j) = min Then

oa += 2 : ob += a1(j) \* 2

If ob = ic(1) Then : GoTo 1

ElseIf ob > ic(1) Then

ob -= a1(j) \* 2 : GoTo 1

End If

a1(j) = -1 : a2(j) = -1

End If

Next

For j = 0 To ic(0) - 1

If a2(j) = min Then

oa += 1 : ob += a2(j)

If ob = ic(1) Then : GoTo 1

ElseIf ob > ic(1) Then

ob -= a2(j) : GoTo 1

End If

a1(j) = -1 : a2(j) = -1

End If

Next

If ob < ic(1) Then min += 0.5 : GoTo 2

1: sw.WriteLine(oa)

Next

sw.Flush() : sw.Close() : End

End Sub

End Class