夏季學院通識計算機程式設計期中考參考解答

7/24/2020

1.

(a) 宣告 **bool** 變數 **b**，**string** 變數 **s1** 和 **s2** (3%)

答:

```
bool b;
string s1;
string s2;
```

(b) 在螢幕顯示一行字，要求使用者輸入一個字串 (3%)

答:

```
Console.WriteLine("enter a string");
```

(c) 自鍵盤讀入一個字串，並將其值存入已宣告之 **string** 變數 **s1** (3%)

答:

```
s1 = Console.ReadLine();
```

(d) 令 **string** 變數 **s2** 之值為 **Hello** (3%)

答:

```
s2 = "Hello";
```

(e) 檢查整數變數 **s1** 和 **s2** 值是否相等，將結果存入 **bool** 變數 **b** (3%)

答:

```
b = (s1 == s2);
```

2.

(a) 將已宣告設值之 **int** 變數 **n**，存入他處已宣告設值的 **int** 變數 **m** 後，以算子 **++** 將變數 **n** 之值加 **1**。兩件工作最好能以一個敘述完成 (3%)

答:

```
m = n++;
```

(b) 宣告 **int** 變數 **r**，並設其值為 **m / 2 * 2**。其中 **m** 為他處已宣告設值的 **int** 變數 (3%)

答:

```
int r = m / 2 * 2;
```

(c) 宣告 **double** 變數 **a, b**，分別設定其初值為 **3.0, 4.0**。宣告 **double** 變數 **c**，並設其值為 $\sqrt{a^2+b^2}$ (3%)

答:

```
double a = 3.0;
double b = 4.0;
double c = Math.Sqrt(a*a +b*b);
```

(d) 宣告 **double** 變數 **x**，並設其值為 **-0.5**。其次宣告 **int** 變數 **sgn**，並利用三元運算子，使變數 **x** 數值大於等於 **0** 時，設定變數 **sgn** 的數值為 **1**，反之則令 **sgn** 值為 **-1** (3%)

答：

```
double x = -0.5;
int sgn = (x >= 0)? 1 : -1;
```

(e) 宣告變數 **path** 為 **string** 型別，並令其值代表 Windows 作業系統下的檔案路徑 **D:\repos\Midterm2020Summer\Problem2** (3%)

答:

```
string path = "D:\\repos\\Midterm2020Summer\\Problem2";
Console.WriteLine(path);
```

3.

(a) 寫一個 **for** 迴圈，計算每年本利和。螢幕輸出截圖如圖 1 (3%)

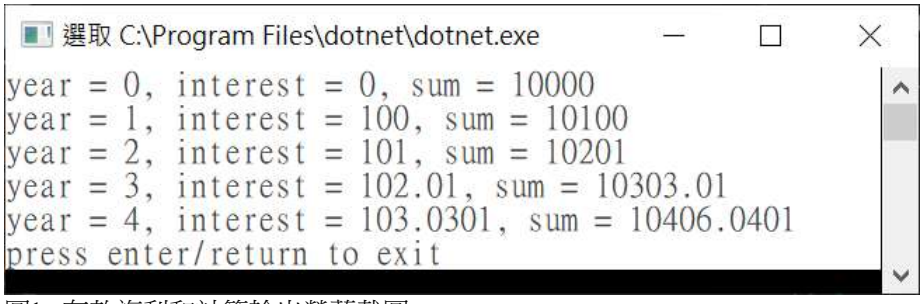


圖1. 存款複利和計算輸出螢幕截圖

答:

```
double sum = 10000;
double rate = 0.01;
double interest = 0.0;
for(int year = 0; year <=4; ++year)
{
   Console.WriteLine(
     "year = {0}, interest = {1}, sum = {2}",
     year, interest, sum);
   interest = sum*rate;
   sum += interest;
}
```

(b) 寫一個 **while** 迴圈，完成(a)小題要求。螢幕輸出截圖也如圖 1 所示

(3%)

答:

```
double sum = 10000;
double rate = 0.01;
double interest = 0.0;;
int year = 0;
while(year <=4)
{
  Console.WriteLine(
    "year = {0}, interest = {1}, sum = {2}",
    year, interest, sum);
  interest = (sum * rate);
  sum += interest;
  ++year;
}
```

(c) 寫一個 **do while** 迴圈，完成(a)小題要求。螢幕輸出截圖也如圖 1 所示 (3 %)

答:

```
double sum = 10000;
double rate = 0.01;
double interest = 0.0;
int year = 0;
do
{
  Console.WriteLine(
    "year = {0}, interest = {1}, sum = {2}",
    year, interest, sum);
  interest = sum * rate;
  sum += interest;
  ++year;
} while(year <= 4);
```

(d) 在(a)小題的 **for** 迴圈中，加入一個條件敘述，使在第二年時，以 **continue** 敘述，跳過當年利息及本利和的計算。螢幕輸出截圖如圖 2 (3%)

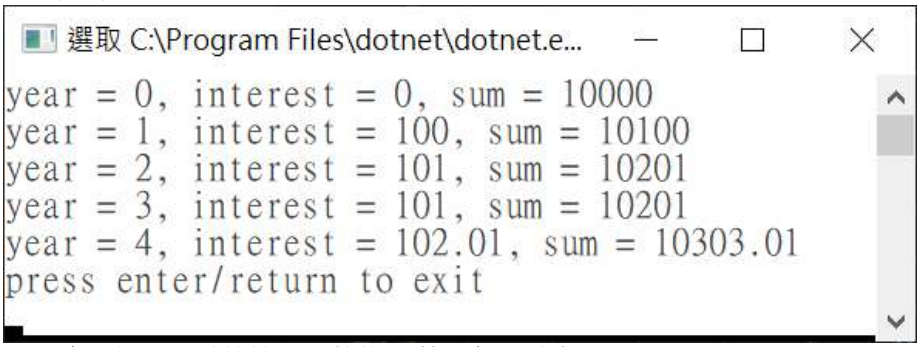


圖2. 存款複利和計算輸出螢幕截圖(第二年不計息)

答:

```
double sum = 10000;
double rate = 0.01;
double interest = 0.0;
for(int year = 0; year <= 4; ++year)
{
  Console.WriteLine(
```

```
    "year = {0}, interest = {1}, sum = {2}",
    year, interest, sum);
  if(year == 2) continue;
  interest = sum*rate;
  sum += interest;
}
```

(e) 在(a)小題的 **for** 迴圈中，加入一個條件敘述，使在第二年時，以 **break** 敘述，跳出迴圈，中止存款。螢幕輸出截圖如圖 3 (3%)

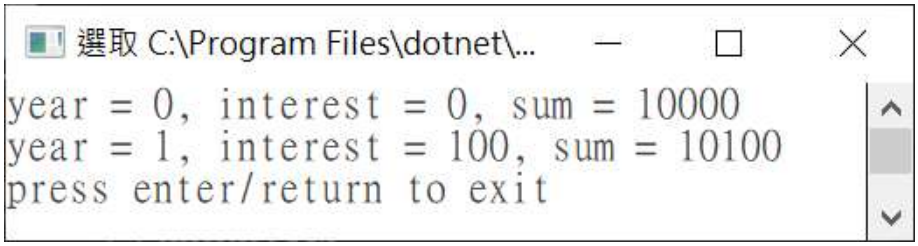


圖3 存款複利和計算輸出螢幕截圖(第二年提前解約，中止存款)

答:

```
double sum = 10000;
double rate = 0.01;
double interest = 0.0;
for(int year = 0; year <= 4; ++year)
{
  if(year == 2) break;
  Console.WriteLine(
    "year = {0}, interest = {1}, sum = {2}",
    year, interest, sum);
    interest = sum*rate;
    sum += interest;
}
```

4.

(a) (3%) (一個語法錯誤)

```
int x;

++x;
```

答:

錯誤:

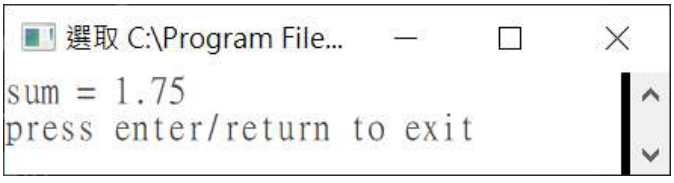
x 沒有設值就進行遞增的計算。

改正:

```
int x = 0;  //can be other value

++x;
```

(b) (3%) (一個語義錯誤) 執行時螢幕應顯示



```
double sum = 1 + 1/2 + 1/4;
Console.WriteLine("sum = " + sum);
```

答:

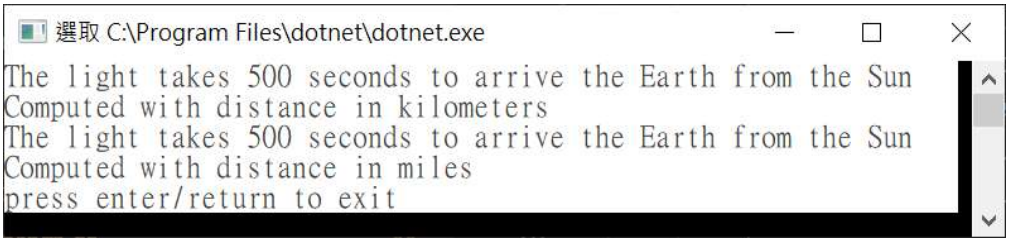
錯誤:

1/2 及 1/4 分別得到 1 除以 2 及除以 4 的商，亦即都是 0。所以 算出的 sum 為 1。為使 sum 的值為 1.75，1/2 及 1/4 要改成浮點數的計算。

改正:

```
double sum = 1 + 1.0/2.0 + 1.0/4.0;
Console.WriteLine("sum = " + sum);
```

(c) (3%) (一個語法錯誤) 執行時螢幕應顯示




```
const double c = 3.0e5; //speed of light, kilometers per second
double distanceFromSunToEarth = 150000000; //kilometers
double time = distanceFromSunToEarth/c; //time for light
                                        //propagating from the
                                        //Sun to the Earth
Console.WriteLine(
"The light takes {0} seconds to arrive the Earth from the Sun",time);
Console.WriteLine("Computed with distance in kilometers");
double kilometersPerMile = 1.609344;
c /= kilometersPerMile;  //light speed, miles per second
distanceFromSunToEarth /= kilometersPerMile; //in miles
time = distanceFromSunToEarth/c;
Console.WriteLine(
"The light takes {0} seconds to arrive the Earth from the Sun",time);
Console.WriteLine("Computed with distance in miles");
```

答:

錯誤:

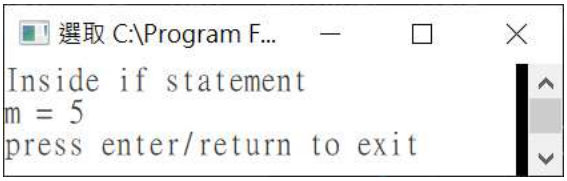
c 宣告為常數後，不可以用 c /= kilometersPerMile; 修改其數值。

改正:

```
const double C = 3.0e5;  //speed of light, kilometers per second
double distanceFromSunToEarth = 150000000; // kilometers
double time = distanceFromSunToEarth/C;
                                   //time for light from sun to earth
Console.WriteLine(
"The light takes {0} seconds to arrive the Earth from the Sun",time);
Console.WriteLine("Computed with distance in kilometers");

const double KM_PER_MILE = 1.609344;
const double C_MI_PS = C/KM_PER_MILE;  // miles per second
distanceFromSunToEarth /= KM_PER_MILE; // miles
time = distanceFromSunToEarth/C_MI_PS;
Console.WriteLine(
"The light takes {0} seconds to arrive the Earth from the Sun",time);
Console.WriteLine("Computed with distance in miles");
```

(d) (3%) (一個語義錯誤) 執行時螢幕應顯示如下:




```
int n = 5;
int m = 4;
if(n != n/2*2 || (++m) < 6)
{
   Console.WriteLine("Inside if statement");
}
Console.WriteLine("m = " + m);
```

答:

錯誤:

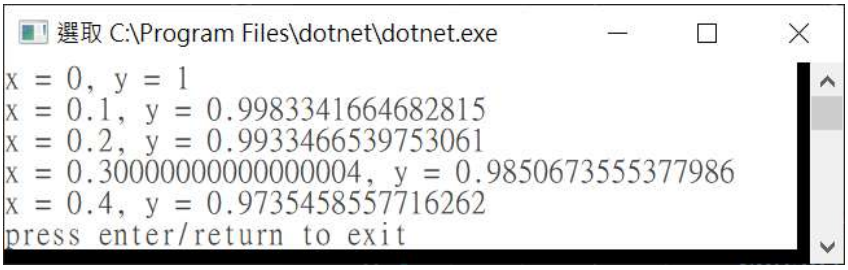
由於 n = 5，n/2 = 2，因此 n/2*2 = 4，不等於 5，所以 n != n/2*2 為真，使整個邏輯條件為真。由於 || 算子代表邏輯「或」的 short circuit 運算，因此不必執行 (++m) < 6 ，就進入 if 大括弧內，寫出 Inside if statement，最後顯示 m = 4，與所求不同。必須將 || 算子改為 |，不採 short circuit 運算，使 (++m) < 6 也會被執行。

改正:

```
int n = 5;
int m = 4;
```

```
if(n != n/2*2 | (++m) < 6)
{
   Console.WriteLine("Inside if statement");
}
Console.WriteLine("m = " + m);
```

(e) (3%) (一個語義錯誤) 執行時螢幕應顯示如下:




```
double deltax = 0.1;
double x;
double y;
int N = 5;
for(int i = 0; i < N; ++i )
{
  x = i*deltax;
  y = Math.Sin(x)/x;
  Console.WriteLine("x = {0}, y = {1}", x, y );
}
```

提示: 數學上利用極限觀念，可以令 $x = 0$ 時的 $\frac{\sin}{x}$ 等於 1。所以本題應檢查每個迴圈控制變數 **i**：如果對應的 **x** 絕對值**(Math.Abs(x))** 小於某個門檻值(例如10^{-6})時，跳過計算，直接設 **y** 的值為 **1**。這可以避免除以零，造成電腦執行的錯誤。

答:

錯誤:

迴圈控制變數 i 為 0 時，Math.Sin(x) 必須除以零，造成電腦執行錯誤。

改正:

```
double deltax = 0.1;
double x;
double y;
int N = 5;
for(int i = 0; i < N; ++i)
```

```
    {
      x = i*deltax;
      y = (Math.Abs(x) < 1.0e-6) ? 1.0 : Math.Sin(x)/x;
      Console.WriteLine("x = {0}, y = {1}", x, y );
    }
```

5. 試寫出下列程式的螢幕輸出。假設使用者在程式第一次提示輸入 **m** 值時，鍵入 **13**，而程式第二次要求輸入時，鍵入 **7**。 (10 %)

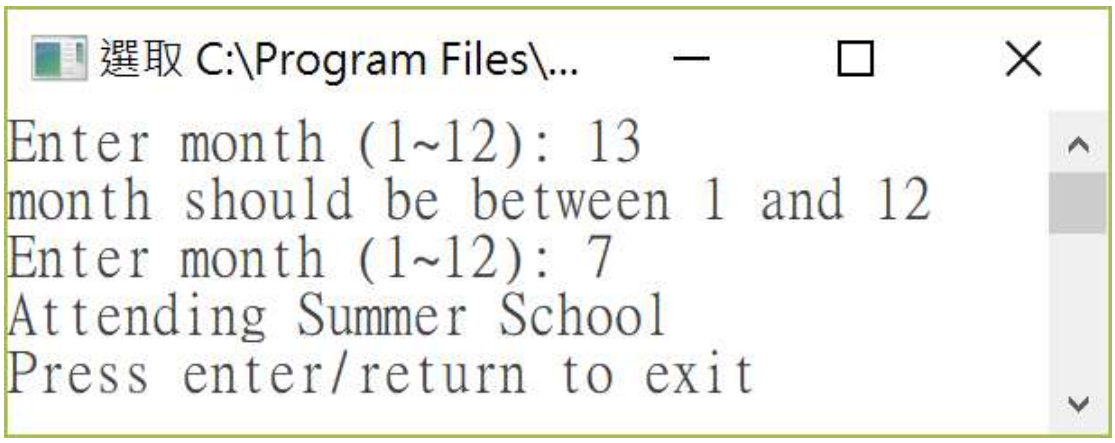
```
using System;

namespace Problem5
{
    enum Season
    {
        Spring,
        Summer,
        Fall,
        Winter
    }
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter month (1~12): ");
            int m = int.Parse(Console.ReadLine());
            while( m < 1 || m > 12)
            {
                Console.WriteLine("month should be between 1 and 12");
                Console.Write("Enter month (1~12): ");
                m = int.Parse(Console.ReadLine());
            }

            Season season;
            if(m == 12 || m == 1 || m == 2)
            {
                season = Season.Winter;
            }
```

```
            if(m >= 3 && m <= 5)
            {
                season = Season.Spring;
            }
            else if(m >= 6 && m <= 8)
            {
                season = Season.Summer;
            }
            else
            {
                season = Season.Fall;
            }
            switch(season)
            {
                case Season.Spring:
                    Console.WriteLine("Attending Spring semester");
                    break;
                case Season.Summer:
                    Console.WriteLine("Attending Summer College");
                    break;
                case Season.Fall:
                    Console.WriteLine("Attending Fall semester");
                    break;
                case Season.Winter:
                    Console.WriteLine("Taking Winter vacation");
                    break;
                default:
                    Console.WriteLine("Should not be here");
                    break;
            }
            Console.WriteLine("Press enter/return to exit");
            Console.ReadLine();
        }
    }
}
```

答:

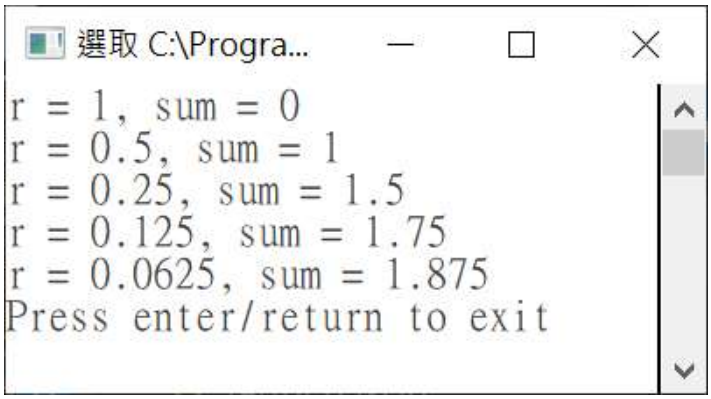


6. 試寫出下列程式的螢幕輸出 (5 %)

```
using System;

namespace Problem6
{
    class Program
    {
        static void Main(string[] args)
        {
            double r = 1.0;
            double sum = 0.0;
            for(int i = 0; i <5; ++i)
            {
                Console.WriteLine("r = {0}, sum = {1}", r, sum);
                sum += r;
                r *= 0.5;
            }
            Console.WriteLine("Press enter/return to exit");
            Console.ReadLine();
        }
    }
}
```

答:



7. Zeno 悖論(https://en.wikipedia.org/wiki/Zeno%27s_paradoxes)是古希臘哲學家 Zeno 提出的論述：假設 Achilles (Homer 史詩 Iliad 中的英雄人物。Iliad 描述 Troy 戰爭，即《木馬屠城記》的過程)與一隻烏龜賽跑，只要烏龜先爬一段距離，Achilles 就永遠追不上烏龜。參考圖 4，「證明」如下：假定(超級)烏龜每單位時間爬行 1 公尺，Achilles 每單位時間跑 10 公尺。令烏龜先前進 100 公尺，則 Achilles 需要跑 10 單位時間，抵達烏龜的原先位置。但這段時間烏龜已經又前進了 10 公尺。因此 Achilles 再花 1 單位時間，跑到烏龜的上一次位置。然而，在這 1 單位時間中，烏龜再度往前移動了 1 公尺。如此，每次 Achilles 抵達烏龜的前一時間位置，烏龜就又往前挪一段距離(雖然這領先距離越來越短)，所以「Achilles 永遠追不上烏龜」。請你寫一個完整程式，模仿上述的推論過程，直到兩者的距離小於等於10^{-4}公尺為止。輸出螢幕的截圖如圖 5 所示，注意其中顯示的數值捨去誤差(round off error)約為10^{-15}，遠低於我們所設的門檻10^{-4}，可以忽略不計。

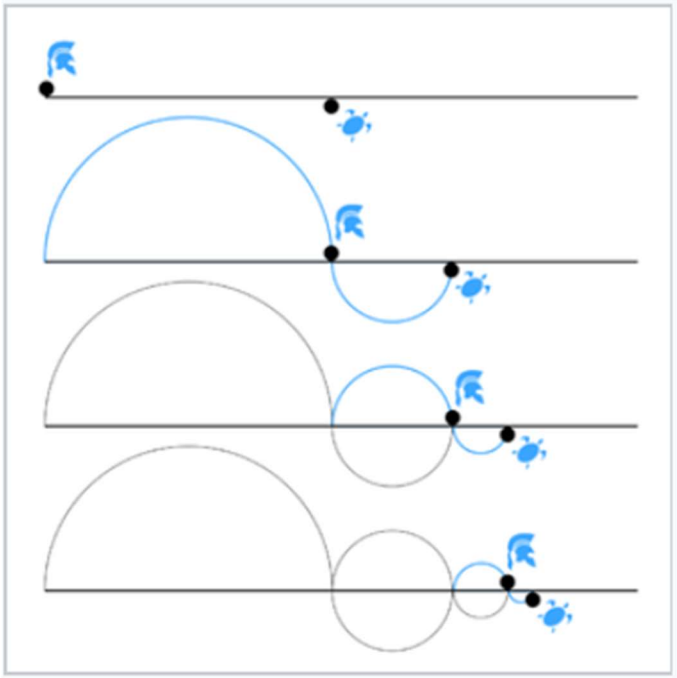


圖4. Zeno 悖論圖解.
取自 https://en.wikipedia.org/wiki/Zeno%27s_paradoxes

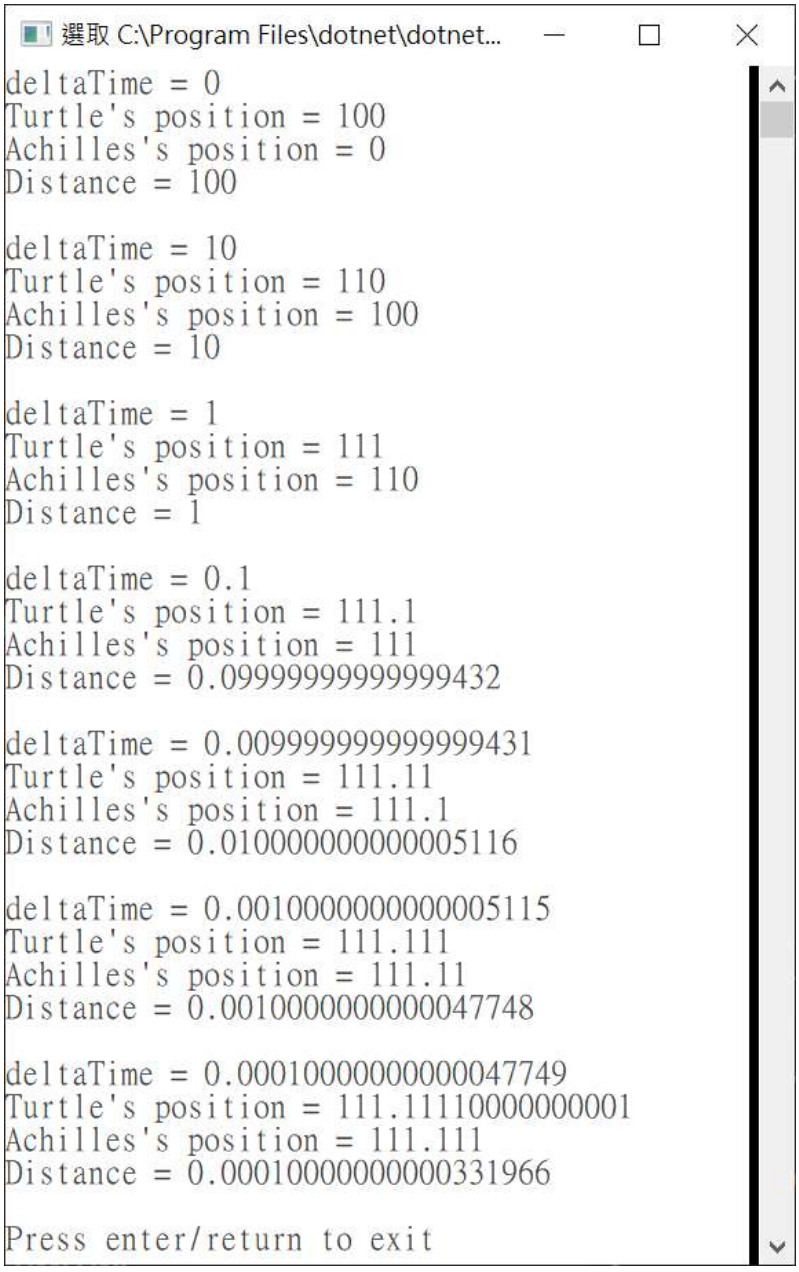


圖5. Zeno 悖論模擬輸出畫面

本題滿分 25 分。(25%)

答:

```
using System;

namespace Problem7
{
    class Program
    {
        static void Main(string[] args)
        {
            double turtlePosition = 100.0;
            double achillesPosition = 0.0;
```

```
            double distance = turtlePosition - achillesPosition;
            double turtleSpeed = 1.0;
            double achillesSpeed = 10.0;
            double deltaTime = 0;
            do {
               Console.WriteLine(
                  "deltaTime = " + deltaTime);
               Console.WriteLine(
                  "Turtle's position = " + turtlePosition);
               Console.WriteLine(
                  "Achilles's position = " + achillesPosition);
               Console.WriteLine("Distance = " + distance);
               Console.WriteLine();
               deltaTime = distance / achillesSpeed;
               achillesPosition = turtlePosition;
               turtlePosition += turtleSpeed * deltaTime;
               distance = turtlePosition - achillesPosition;
            } while(Math.Abs(distance) > 1.0e-4);

            Console.WriteLine("Press enter/return to exit");
            Console.ReadLine();
         }
      }
   }
```