

程式設計 作業4			
系級	航空與太空工程學系		
學號	F44134057	姓名	邱翊碩

1. 寫一程式，輸入秒數，並輸出小時/分/秒。(20%)

.cpp



執行結果

Microsoft Visual Studio 偵錯主控台

1564856  
434小時/40分/56秒  
C:\Users\eamon\OneDrive\桌面\Mywork\程式作業\第四次作業\HW4\_Q1\Debug\HW4\_Q1.exe (處理序 37996) 已結束，代碼為 0。  
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [在偵錯停止時自動關閉主控台]。  
按任意鍵關閉此視窗 . . .

Microsoft Visual Studio 偵錯主控台

500  
0小時/8分/20秒  
C:\Users\eamon\OneDrive\桌面\Mywork\程式作業\第四次作業\HW4\_Q1\Debug\HW4\_Q1.exe (處理序 39120) 已結束，代碼為 0。  
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [在偵錯停止時自動關閉主控台]。  
按任意鍵關閉此視窗 . . .

Microsoft Visual Studio 偵錯主控台

36  
0小時/0分/36秒  
C:\Users\eamon\OneDrive\桌面\Mywork\程式作業\第四次作業\HW4\_Q1\Debug\HW4\_Q1.exe (處理序 16204) 已結束，代碼為 0。  
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [在偵錯停止時自動關閉主控台]。  
按任意鍵關閉此視窗 . . .

2. 說明函式 getch()、getche()與 getchar()的差異。(20%)

三者都是用來輸入字元，其中差異如下：

getch()在輸入字元之後會直接從鍵盤緩衝區讀進，所以不需要按下 Enter 健，而且由 getch()所接收的字元不會顯示在螢幕上，剛剛打的字元會有消失的感覺。

getche()在輸入字元之後也會直接從鍵盤緩衝區讀進，所以不須按 Enter，與 getch()不同的是 getche()輸入的字元會顯示在螢幕上不會消失。

getchar()在輸入字元之後會進入鍵盤緩衝區，按下 Enter 之後才會完成整個輸入，這點與以上兩者不同，且由 getchar()輸入的字元也會持續留在螢幕上。

3. (程式片段)

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int num;
    printf("輸入一整數:");
    scanf("%d",num);
    printf("整數=%d \n", num);
    system("pause");
    return 0;
}
```

程式執行時發生了錯誤，請問哪裡需要更正？(20%)

函式 scanf()的第二個參數應輸入記憶體位置，但上面的程式在使用 scanf()時填入的第二個變數輸入的是變數的數值，應該要加上&取址運算址，改成&num 之後輸入函式本身才能順利執行。

4. 寫一程式，輸入圓形的半徑，印出其面積(需定義 pi)。(20%)

.cpp

```
1 #include <stdio.h>
2 #include <math.h> //使用acos(-1)作為圓周率
3
4 int main()
5 {
6     float r; //宣告長度
7     double pi = acos(-1); //圓周率的值
8     scanf_s("%f", &r); //輸入半徑
9     printf("%f", pi * r * r); //直接輸出面積
10    return 0;
11 }
```

執行結果

Microsoft Visual Studio 偵錯主控台

3  
28.274334  
C:\Users\eamson\OneDrive\桌面\Mywork\程式作業\第四次作業\HW4\_Q4\Debug\HW4\_Q4.exe (處理序 16412) 已結束，代碼為 0。  
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [在偵錯停止時自動關閉主控台]。  
按任意鍵關閉此視窗 . . .

Microsoft Visual Studio 偵錯主控台

123  
47529.155256  
C:\Users\eamson\OneDrive\桌面\Mywork\程式作業\第四次作業\HW4\_Q4\Debug\HW4\_Q4.exe (處理序 22228) 已結束，代碼為 0。  
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [在偵錯停止時自動關閉主控台]。  
按任意鍵關閉此視窗 . . .

Microsoft Visual Studio 偵錯主控台

5.5  
95.033178  
C:\Users\eamson\OneDrive\桌面\Mywork\程式作業\第四次作業\HW4\_Q4\Debug\HW4\_Q4.exe (處理序 26252) 已結束，代碼為 0。  
若要在偵錯停止時自動關閉主控台，請啟用 [工具] -> [選項] -> [偵錯] -> [在偵錯停止時自動關閉主控台]。  
按任意鍵關閉此視窗 . . .

5. 依照以下程式碼，分別說明 5 項輸出 a 之間的差異。(20%)

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
```

```
float a = 1234.5678;
printf("a=%f\n", a);
printf("a=%.f\n", a);
printf("a=%.3f\n", a);
printf("a=%5.2f\n", a);
printf("a=%7.2f\n", a);
system("pause");
return 0;
}
```

第一個 a 會印出 1234.567800，%f 可以輸出有號浮點數，且%f 的輸出預設精度是小數點後六位，就算小數點不滿六位一樣會用 0 補滿，所以會在 8 後面補上兩個 0。

第二個 a 會印出 1235，在%後面加上一個小數點代表等等直接忽略小數點並且在個位數四捨五入，小數點後第一位是 5，所以個位數 4 會變成 5。

第三個 a 會印出 1234.568，在%後面加上.3 代表輸出整數部分後在小數點後三位的四捨五入，小數點後三位原本是 7，但第四位是 8，所以四捨五入 7 變成 8。

第四個 a 會印出 1234.57，%5.2 的 5 代表預留五個空格，如果數字的寬度(包含小數點)不超過五格，會以空格填在前面，但如果數字的寬度大於原本預留的空間的話就會直接照樣輸出，逗點後的 2 則表示輸出的數值保留到小數點後二位，在小數點後二位四捨五入 6 變成 7。

第五個 a 會印出 1234.57，%7.2 表示預留 7 個空格，但數字輸出到小數點後第二位，已經(加上逗點)總寬度已經達到了 7 格，所以不會也不會特別在前面加上空白鍵，%.2 如上，輸出到小數點第二位，並在小數點第二位四捨五入。