

1061 NTOUCSE 程式設計 1C 期中考 參考答案

姓名：_____ 系級：_____ 學號：_____

106/11/07 (二)

考試時間：14:10 – 16:00

1. [10] 請問 C 程式中 double 資料型態的變數以什麼格式存放在記憶體裡？大約的範圍是？(請以 10 的次方表示) 一個實際世界中大小在 10^{200} 的數值要以 double 型態存放時出現的誤差大約是多少？(請以 10 的次方表示) 程式執行時 sizeof(double) 的數值為多少？使用 scanf 由鍵盤讀入 double 型態變數時需要使用什麼格式轉換命令？

Sol:

IEEE754 的倍精準浮點數格式

請參考 http://www.binaryconvert.com/convert_double.html

能夠存放的數字範圍大約在 $\pm 10^{-308} \sim 10^{308}$

大約 $10^{185} \sim 10^{186}$

sizeof(double)的數值是 8

%lf

2. [10] 請問下面程式執行後列印出來的結果是？

```
int s = 1;
void add(int a)
{
    int s = 6;
    for (; a>=0; --a)
    {
        printf("%d,", s);
        s++;
        printf("%d,", s);
    }
}
int main()
{
    printf("%d,", s);
    add(s);
    printf("%d,", s);
    s = 9;
    printf("%d", s);
    return 0;
}
```

Sol:

1,6,7,7,8,1,9

3. [10] 有一個同學想要寫一個程式模擬隨機丟骰子，他希望丟的次數有一半的機會是 5 次另外一半的機會是 6 次，所以就寫了下面的這段程式

```
int i;
for (i=0; i!=rand()%2+5; i++)
    printf("%d ", rand()%6+1);
```

請問這段程式執行時有多大機會印出 5 或 6 個數字? (假設 `rand()%2` 為 0 和為 1 的機會各為 1/2), 該怎麼修改讓它符合需求?

Sol:

$$1/2 + 1/2 * 1/2 = 3/4$$

```
int i, k;
k= rand()%2+5;
for (i=0; i<k; i++)
    printf("%d ", rand()%6+1);
```

```
int i;
for (i=0; i<rand()%2+5; i++)
    printf("%d ", rand()%6+1);
```

也剛剛好可以達成需求,但是概念上是「到達 5 次以後每一次丟銅板決定要不要繼續」,而且在迴圈控制的地方前五次丟銅板的動作是白作的,和前一個程式模擬隨機決定 5 次或是 6 次是不一樣的

4. [10] 有同學想要檢查輸入的數字是不是在 0 到 999 之間(包含 0 和 999),他寫了下面的程式

```
int n;
do
{
    printf("Please input a number in [0,999]: ");
    scanf("%d", &n);
}
while (n>999, n<0);
```

這段程式可以正確編譯,但是執行起來怪怪的,輸入 -1 的時候會要求重複輸入,但是輸入 1234 的時候不會,請解釋會有這個表現的原因? 請問該如何修改?

Sol:

`n>999, n<0` 由運算子「`,`」,「`>`」,「`<`」以及運算元「`n`」,「`999`」,「`0`」組成,是一個語法正確的運算式(expression),執行的時候由左邊 `n>999` 開始做,得到 0 或是 1,但是這個結果會丟棄不用,接下來執行 `n<0`,得到 0 或是 1,當作是整個運算式的結果,所以由鍵盤輸入數字 -1 時會得到 1,使得迴圈重複執行一次,由鍵盤輸入 1234 時會得到 0,使得迴圈結束
應該要改成 `while (n>999 || n<0);`

5. [15] 有同學想要寫一個程式把攝氏溫度轉換為華氏溫度,關鍵的程式碼如下:

```
int celcius, fahrenheit;
scanf("%d", &celcius);
fahrenheit = celcius / 5 * 9 + 32;
printf("華氏溫度為 %d 度\n", fahrenheit);
```

他發現輸入攝氏 10 度的時候印出「華氏溫度為 50 度」,覺得挺滿意的,但是給別人測試一下的時候發現輸入其它溫度好像都怪怪的,你能夠預測一下輸入哪些溫度時轉換結果會是對的? 請問程式該怎麼改才能使得轉換的誤差在華氏 1 度以下? 請問怎樣改才能夠讓轉換的誤差不超過華氏 0.5 度?

Sol:

..., -10, -5, 0, 5, 10, 15, 20, 25, ... 5 的倍數都會是正確的

```
fahrenheit = (int)(celcius / 5.0 * 9) + 32;  
或是 fahrenheit = celcius / 5.0 * 9 + 32;  
或是 fahrenheit = (double)celcius / 5 * 9 + 32;  
或是其它組合
```

```
fahrenheit = (int)(celcius / 5.0 * 9 + 0.5) + 32;  
或是 fahrenheit = celcius / 5.0 * 9 + 32.5;  
或是其它組合
```

如果改成 `fahrenheit = celcius * 9 / 5 + 32;`

雖然也會得到誤差在 1 以下的華氏溫度，仔細推敲誤差會在 0.8 以內，不過程式的目的在運用正確的語法撰寫一個正確的轉換工具，這樣的程式不管怎樣都是用錯誤的方法使用程式裡表達資料的型態系統，後面其實也沒有辦法用正常的四捨五入功能控制誤差在 0.5 以內

6. [10] 有一個程式希望執行程式的人先輸入年齡，然後再輸入身份證號碼(台灣的身份證字號類似 A123456789)，程式裡這樣子寫：

```
char ch;  
int age, number;  
printf("Please input your age: ");  
scanf("%d", &age);  
printf("Please input your identification number: ");  
scanf("%c%d", &ch, &number);  
printf("age:%d ID:%c%d\n", age, ch, number);
```

程式執行的時候看到下面的結果：

```
Please input your age: 19  
Please input your identification number: A123456789  
age:19 ID:  
0
```

請解釋為什麼會看到這種結果？該如何修改讓結果正確地印出來？

Sol:

因為輸入 19 以後輸入串流裡還有<enter>，所以讀到 ch 裡面的是'\n'，在印出 ch 時會以換列的方式呈現，由於 `scanf("%c%d",&ch,&number);` 完成讀取 ch 字元以後，在輸入串流裡看到 A 這個非 0~9 的資料，因此並沒有把資料讀入 number 變數中，印出的 0 是先前的程式剛剛好留在 number 變數記憶體裡面的數字

要處理這個問題常常可以用下面幾種方式來處理

方法一：`scanf("□%c%d", &ch, &number);` // 以空格命令來跳過所有 white space

方法二：`scanf("%c%c%d", &ch, &ch, &number);` // 以額外的 %c 讀入 '\n'

方法三：`scanf("%*c%c%d", &ch, &number);` // %*c, 基本上和 %c 一樣，但是讀進來的東西不存到記憶體裡面

方法四：`getchar();` // 讀入單一字元，不存放到任何變數裡

`scanf("%c%d", &ch, &number);`

方法五：`char buf[100];`

`fgets(buf, 100, stdin);` // 或是 `gets(buf);` 如果擔心在 19 之後不只一個 <enter>

`scanf("%c%d", &ch, &number);`

方法六：`fflush(stdin);` // 各系統表現不一，不建議使用

`scanf("%c%d", &ch, &number);`

7. [10] 若 `A[i][j]` 是一個 10×50 的二維整數陣列，裡面存放了 500 個大於 0 的整數，下列函式希望計算 A 陣列每一列的總和，請問列印出來的 10 列敘述哪些是不正確的？該怎樣修改？

```
void sum(int A[10][50])
{
    int i, j, rowsum = 0;
    for (i=0; i<10; i++)
    {
        for (j=0; j<50; j++)
            rowsum = rowsum + A[i][j];
        printf("The sum of row %d is %d.\n", i, rowsum);
    }
}
```

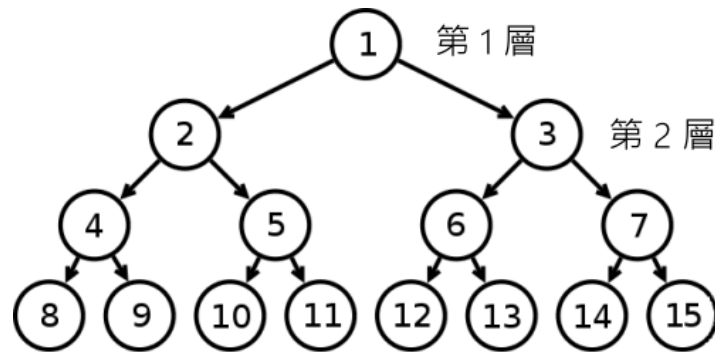
Sol:

除了第一列之外其他的輸出都是錯的

需要改成

```
int i, j, rowsum;
for (i=0; i<10; i++)
{
    rowsum = 0;
    for (j=0; j<50; j++)
        rowsum = rowsum + A[i][j];
    printf("The sum of row %d is %d.\n", i, rowsum);
}
```

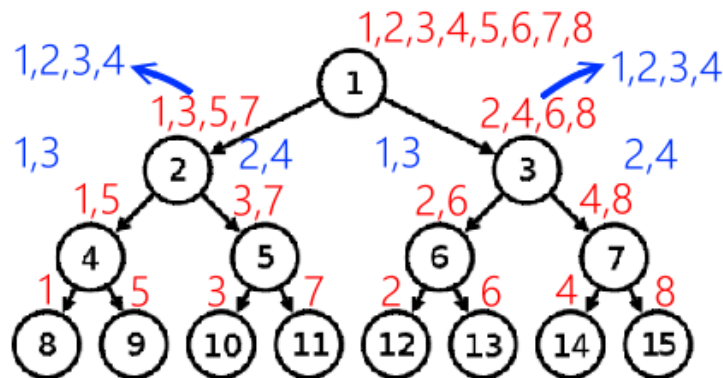
8. [10] 如下圖有一個彈珠台軌道安排成 $L(=4)$ 層的架構，彈珠由第 1 層標示為 1 的節點滾下，一直滾到第 L 層才會停下，第 L 層有 2^{L-1} 個節點，編號為 2^{L-1} 到 2^L-1 ，節點 1 到節點 $2^{L-1}-1$ 是有記憶的分流裝置，如果前一顆經過此節點的彈珠往左(右)下方節點滾過去，下一顆經過此節點的彈珠就會往右(左)下方節點滾過去，一開始的時候每一個分流裝置都開向左，順序滾下第 1 顆球、第 2 顆球、...，一直到第 $K(\leq 2^{L-1})$ 顆球，請完成下列程式，由鍵盤讀入 L 與 K，計算第 K 顆球停在第 L 層編號多少的節點上？



```
int L, K, i, j;
scanf("%d%d", _____);
j = 1;
for (i=1; i<L; i++)
{
    j = _____*j + _____;
    K = _____ ? _____ : K/2;
}
printf("%d\n", j);
```

提示：第 i 層編號 j 的節點的左下方節點編號為 $2j$ ，右下方節點編號為 $2j+1$

Sol:



上圖中紅字是球的編號以及它會走到哪一個節點，基本的規則是每一個節點看到的球的序列中奇數項往左，偶數項往右，所以在 1 號節點的地方公式很容易寫，奇數號碼的球就是序列裡奇數順序的球，所以我們用 $K\%2$ 來判斷奇數偶數，球目前在第 j 個節點，如果是 K 是奇數下一個到達的節點就是 $2*j+1$ ，如果是 K 是偶數下一個到達的節點就是 $2*j$ ；到了 2 號節點的地方就比較討厭，1,3,5,7 都是奇數，規則需要修改，不過如果我們能夠把這些數字轉為藍字的 1,2,3,4，就可沿用原先在 1 號節點的規則了，這個轉換其實也很容易算，就是 $(K+1)/2$ ，3 號節點的地方也需要類似的轉換，不過更簡單了，就是 $K/2$

```
int L, K, i, j;
scanf("%d%d", &L, &K);
j = 1;
for (i=1; i<L; i++)
{
```

```

    j = 2*j + (1-K%2);
    K = K%2 ? (K+1)/2 : K/2;
}
printf("%d\n", j);

```

9. [15] 令一個十六進位(以 0123456789ABCDEF 十六個符號表示)數字的奇數位數和為 A，偶數位數和為 B，請由鍵盤讀入一個十六進位數字(一定符合規則，最多 1000 位數)，在螢幕上以二進位印出 A 與 B 的差的絕對值 |A-B|。例如：輸入 26A54C<enter> 時奇數位數的和 A = 6+5+12 = 23，偶數位數的和 B = 2+10+4 = 16，|A-B| = |23-16| = 7 (二進位表示法為 111)。

Sol:

```

#include <stdio.h> // scanf, printf
int main()
{
    int odd=1, value=0, weight;
    char ch;
    while (scanf("%c", &ch), ch!='\n') // 以 <enter> 結束
    {
        if ((ch>='0')&&(ch<='9'))
            ch = ch-'0';
        else
            ch = ch-'A'+10;
        if (odd)
            value = value + ch;
        else
            value = value - ch;
        odd = 1 - odd;
    }
    value = value>=0 ? value : -value;

    weight = 1;
    while (value >= weight)
        weight *= 2;
    if (weight>1) weight /= 2;
    while (weight > 0)
    {
        printf("%d", value/weight);
        value %= weight;
        weight /= 2;
    }
    printf("\n");
    return 0;
}

```

或是

```

#include <stdio.h> // scanf, sscanf, printf
int main()
{

```

```

int x, i=0, value=0, weight;
char buf[1001];
scanf("%s", buf);
while (1==sscanf(&buf[i], "%1x", &x)) // 處理 buf 時如果字串結束就結束了
{
    value += (i+1)%2 ? x : -x;
    i++;
}
value = value>=0 ? value : -value;

weight = 1;
while (value >= weight)
    weight *= 2;
if (weight>1) weight /= 2;
while (weight > 0)
{
    printf("%d", value/weight);
    value %= weight;
    weight /= 2;
}
printf("\n");
return 0;
}

```

注意 scanf("%1x",&x) 會跳過所有的 white space, 所以無法以 <enter> 結束, 但是可以用任何非 [0-9A-F\n\t] 的字元結束