國立屏東大學 資訊工程系 程式設計(二)

Turnin作業1

- Turnin Code: c.hw1
- Due Date: 3/11 00:00 **Hard Deadline**

繳交方式說明

本次Turnin作業包含多個程式題，請為每一題建立一個資料夾，並將該題所要上傳的檔案放置其中後，再使用turnin指令上傳作業。請同學先為本次作業建立一個資料夾hw1，然後在hw1裡分別為每一題建立一個子資料夾，用以進行每一題的作答以及上傳。每一題的子資料夾名稱已寫於題目前方，請務必依照題目的規定建立子資料夾，例如第1題為p1，第2題為p2，餘依此類推。當我們完成某一個題目的作答後，就可以使用turnin指令將該題的答案上傳。以第1題為例，當我們在p1子資料夾裡完成作答後，就可以回到hw1資料夾，使用以下指令將其上傳：

[3:23 user@ws hw1] `turnin␣c.hw1␣p1↵`

當然，你也可以等到所有題目都完成後，回到hw1資料夾，使用以下指令將所有題目都加以上傳：

[3:23 user@ws hw1] `turnin␣c.hw1␣.↵`

註：本文使用　及　代表空白字元與Enter換行字元，並且將使用者輸入的部份使用灰階方式顯示。

---

p1 兩數交換函式設計

請參考下面的Main.c程式:

```
#include <stdio.h>
#include "Swap.h"

int main()
{
   int x, y, *p;
   printf("x=?");
   scanf("%d", &x);
   printf("y=?");
   scanf("%d", &y);
   swap(&x, &y);
   printf("x=%d y=%d\n", x, y);
}
```

你必須完成名為Swap.c與Swap.h的C語言程式，其中分別包含swap()函式的Implementation與其Prototype宣告。swap()函式接收兩個數值的記憶體位址做為參數，並在函式中將這兩個記憶體位址內所包含的數值進行交換。本題的相關程式將使用以下的Makefile進行編譯：

```
all: Main.c Swap.o
        cc Main.c Swap.o

Swap.o: Swap.c Swap.h
        cc -c Swap.c

clean:
        rm -f *.o *~ *.*~ a.out
```

此題的執行結果如下：

```
[3:23 user@ws hw] ./a.out↵
x=?1↵
y=?2↵
x=2▲y=1↵
[3:23 user@ws hw] ./a.out↵
x=?3↵
y=?5↵
x=5▲y=3↵
[3:23 user@ws hw]
```

請注意本題應繳交Swap.c及Swap.h兩個檔案，至於Main.c與Makefile則不需繳交。

p2 傳回最小值的記憶體位址函式設計

請參考下面的Main.c程式:

```
#include <stdio.h>
#include "Min.h"

int main()
{
   int x, y, *p;
   scanf("%d %d", &x, &y);
   p = min(&x, &y);
   printf("The minimum of %d and %d is %d.\n", x, y, *p);
}
```

你必須完成名為Min.c與Min.h的C語言程式，其中分別包含min()函式的Implementation與其Prototype宣告。min()函式接收兩個整數的記憶體位址做為參數，並將其記憶體位址中所包含的數值最小值者的記憶體位址傳回。本題的相關程式將使用以下的Makefile進行編譯：

```
all: Main.c Min.o
        cc Main.c Min.o

Min.o: Min.c Min.h
        cc -c Min.c

clean:
        rm -f *.o *~ *.*~ a.out
```

此題的執行結果如下：

[3:23 user@ws hw] ./a.out↵
1▲2↵
The▲minimum▲of▲1▲and▲2▲is▲1.↵
[3:23 user@ws hw] ./a.out↵
19▲12↵
The▲minimum▲of▲19▲and▲12▲is▲12.↵
[3:23 user@ws hw]

請注意本題應繳交Min.c及Min.h兩個檔案，至於Main.c與Makefile則不需繳交。

p3 將浮點數四捨五入到指定位數的函式設計

請參考下面的Main.c程式，完成所需的rounding()函式設計(rounding()函式的prototype及implementation請分別寫在Round.h及Round.c檔案裡。):

```
#include <stdio.h>
#include "Round.h"

int main()
{
    double x;
    int p;
    printf("Please input a floating number: ");
    scanf(" %lf", &x);
    printf("What decimal place do you want to round? ");
    scanf(" %d", &p);
    rounding(&x,p);
    printf("%f\n", x);
}
```

呼叫rounding()函式時必須傳入兩個引數：

- 一個浮點數所在的記憶體位址□double *num

- 一個代表位數的整數值□int pos (本題用以測試的$0\leq$pos$\leq$5)

此函式的作用是將所傳入的記憶體位址內的浮點數數值，四捨五入到小數點後第pos位(假設pos□3 則113.641590 – > 113.64200)的運算，且其運算結果必須存放於該記憶體位址內。

本題的相關程式將使用以下的Makefile進行編譯：

```
all: Main.c Round.o
        cc Main.c Round.o

Round.o: Round.c Round.h
        cc -c Round.c

clean:
        rm -f *.o *~ *.*~ a.out
```

此題的執行結果如下：

[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲floating▲number:▲113.645192↵
What▲decimal▲place▲do▲you▲want▲to▲round?▲0↵
114.000000↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲floating▲number:▲113.645192↵
What▲decimal▲place▲do▲you▲want▲to▲round?▲1↵
113.600000↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲floating▲number:▲113.645192↵
What▲decimal▲place▲do▲you▲want▲to▲round?▲2↵
113.650000↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲floating▲number:▲113.645192↵
What▲decimal▲place▲do▲you▲want▲to▲round?▲3↵
113.645000↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲floating▲number:▲113.645192↵
What▲decimal▲place▲do▲you▲want▲to▲round?▲4↵
113.645200↵
[3:23 user@ws hw]

請注意本題應繳交Round.c及Round.h兩個檔案，至於Main.c與Makefile則不需繳交。

p4 分解浮點數的函式設計

請參考下面的Main.c程式:

```
#include <stdio.h>
#include "Float.h"

int main()
{
   int i_part;
   double f_part;
   double d;
   printf("Please input a floating number: ");
   scanf("%lf", &d);
   decompose(d, &i_part, &f_part);
   printf("i_part=%d\n", i_part);
   printf("f_part=%f\n", f_part);
}
```

你必須完成名為Float.c與Float.h的C語言程式，其中分別包含decompose()函式的Implementation與其Prototype宣告□decompose()函式接收一個浮點數值以及兩個數值的記憶體位址做為參數，並在函式中將所傳入的浮點數的整數與小數部份分別寫入到這兩個記憶體位址內。本題的相關程式將使用以下的Makefile進行編譯：

```
all: Main.c Float.o
        cc Main.c Float.o

Float.o: Float.c Float.h
        cc -c Float.c

clean:
        rm -f *.o *~ *.*~ a.out
```

此題的執行結果如下：

[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲floating▲number:▲3.1415926↵
i_part=3↵
f_part=0.141593↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲floating▲number:▲3↵
i_part=3↵
f_part=0.000000↵
[3:23 user@ws hw] ./a.out↵
Please▲input▲a▲floating▲number:▲0.23↵
i_part=0↵
f_part=0.230000↵
[3:23 user@ws hw]

請注意本題應繳交Float.c及Float.h兩個檔案，至於Main.c與Makefile則不需繳交。

p5 學生成績統計函式設計

請參考下面的Main.c程式:

```
#include <stdio.h>
#include "Score.h"

int main()
{
   int i, max, min;
   double avg;
   int scores[10];

   for(i=0;i<10;i++)
   {
      scanf("%d", &scores[i]);
   }
   avg=aggregation(scores, &max, &min);
   printf("avg=%.2f\n", avg);
   printf("max=%d\n", max);
   printf("min=%d\n", min);
}
```

你必須完成名為Score.c與Score.h的C語言程式，其中分別包含aggregation()函式的Implementation與其Prototype宣告，aggregation()函式接收一個具有10個學生成績的陣列以及分別代表最高分與最低分的記憶體位址做為參數，並在函式中將10個學生成績平均成績、最高分與最低分計算出來，並且將最高分與最低分寫入所傳入的記憶體位址內，最後將平均分數傳回。本題的相關程式將使用以下的Makefile進行編譯：

```
all: Main.c Score.o
	cc Main.c Score.o

Score.o: Score.c Score.h
	cc -c Score.c

clean:
	rm -f *.o *~ *.*~ a.out
```

此題的執行必須輸入10個學生的成績，你可以先備妥輸入用的測試檔案，以縮減輸入的時間。例如：

```
100 90 80 70 60 50 40 30 20 10
```

```
27 31 43 87 8 98 23 77 89 91
```

此題的執行結果可參考如下：

```
[3:23 user@ws hw] ./a.out▲<▲in.1↵
avg=55.00↵
max=100↵
min=10↵
[3:23 user@ws hw] ./a.out▲<▲in.2↵
avg=57.40↵
max=98↵
min=8↵
[3:23 user@ws hw]
```

請注意本題應繳交Score.c及Score.h兩個檔案，至於Main.c與Makefile則不需繳交。