

2024 NCKU Program Design I HW9

```
36 int main(int argc, char *argv[]) {
37     if (argc != 4) {
38         printf("Error input.\n");
39         return 1;
40     }
41
42     double x = atof(argv[1]);
43     int n = atoi(argv[2]);
44     int m = atoi(argv[3]);
```

我們先從命令列接收輸入參數，並將其轉換為對應的數值型態。程式首先檢查參數數量 (argc) 是否為 4，這包括執行檔名以及 3 個數字參數。如果參數不足，程式會輸出錯誤訊息並提前結束，確保不會進行後續的處理。當輸入正確時，程式使用 `atof` 將第一個數字參數轉換為浮點數並指派給變數 `x`，接著將後兩個參數透過 `atoi` 轉換為整數，分別存入變數 `n` 和 `m`，為後續的數值運算做準備。

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  typedef double (*F)(double, int);
```

接著講解 `main` fcn 以外的。這裏定義一個函數指標類型 `F`，指向接受 `(double, int)` 參數並返回 `double` 的函數。後來我有 `import math.h` 來使用 `pow` 函數，因為自己寫的迴圈的輸出貌似和測字的答案不符。

```
7  double power(double x, int n) {
8      return pow(x, n);
9  }
10
11 double multiply(double x, int n) {
12     x = x * n;
13     return x;
14 }
15
16 double divide(double x, int n) {
17     x = x / n;
18     return x;
19 }
```

這三個都是很基本的函數。`Power` 裏使用了 `math.h` 的函數。`Multiply` 和 `divide` 裏直接在局部變數 `x` 作修改並回傳。

```

21 double powerpower(F func, double x, int n, int m) {
22     double res = pow(func(x, n), m);
23     printf("%.6lf\n", res);
24 }

```

```

46 powerpower(power,x,n,m);
47 powerpower(multiply,x,n,m);
48 powerpower(divide,x,n,m);

```

在 3 個基本運算函數之後，為一個 powerpower 的函數，該函數接收一個函數指標 func 和三個參數 x、n 和 m。它以 func(x, n) 的結果為底數，m 為指數，使用 pow 算次方，最後用 .6lf 輸出能顯示小數點後 6 位的數字。

在 main 函數中呼叫 powerpower 時，可以傳入不同的函數指標 power、multiply 或 divide，使其靈活地執行不同的數學運算，實現 $(x^n)^m$ 、 $(x \times n)^m$ 或 $(x \div n)^m$ 的計算。

```

26 double powermultiply(F func, double x, int n, int m) {
27     x = func(x,n) * m;
28     printf("%.6lf\n", x);
29 }
30
31 double powerdivide(F func, double x, int n, int m) {
32     x = func(x,n) / m;
33     printf("%.6lf\n", x);
34 }

```

```

49 powermultiply(power,x,n,m);
50 powermultiply(multiply,x,n,m);
51 powermultiply(divide,x,n,m);
52 powerdivide(power,x,n,m);
53 powerdivide(multiply,x,n,m);
54 powerdivide(divide,x,n,m);

```

Powermultiply 和 powerdivide 函數也是類似的結構。在這裏我沒有另外開變數來儲存 func(x, n)，因為在運算 $(x^n) * m$ 、 $(x * n) * m$ 、 $(x/n) * m$ 、 $(x^n) / m$ 、 $(x * n) / m$ 和 $(x/n) / m$ 的時候只需要使用一次 func(x, n)。同樣也是在局部變數 x 作修改並回傳。