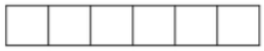
4. 請依照下列要求 寫出部分程式碼

a) 以 malloc 配置一個含 6 個浮點數(float)之 1 維陣列;
並寫**一個迴圈**將此 6 個浮點數設為 {1.1, 1.2, 1.3, 1.4, 1.5, 1.6}



b) 以 malloc 配置一個含 6 個浮點數(float)之 2 維陣列 ([2*3]);
並寫一個**雙層巢狀迴圈** 將此陣列的浮點數設為 {1.1, 1.2, 1.3, 1.4, 1.5, 1.6}

4.

a

```
ptd = (float*) malloc(6*sizeof(float)); //用malloc宣告6個float的空間，將資料型態轉成float*再存入ptd
for(int i=0;i<6;i++)
    ptd[i] = ((float)(i+1)/10) +1; //從0~5，分別為0.1+1, 0.2+1, 0.3+1, 0.4+1, 0.5+1, 0.6+1
```

b

```
ptd = (float**)malloc(2*sizeof(void *)); //用malloc宣告2個void*的空間，將資料型態轉成float**再存入ptd，ptd用來存兩個一維陣列的起始位址，以形成一個二維陣列，此處sizeof(void*)用來表示一個指標的大小，指標所存的內容為記憶體位址，所以此大小會依硬體不同而改變，例如在64位元電腦sizeof(void*)就是8個bytes
ptd[0] = (float*)malloc(3*sizeof(float)); //用malloc宣告3個float的空間，將資料型態轉成float*再存入ptd[0]
ptd[1] = (float*)malloc(3*sizeof(float)); //用malloc宣告3個float的空間，將資料型態轉成float*再存入ptd[1]
for(int i=0;i<2;i++)
    for(int j=0;j<3;j++)
        ptd[i][j] = ((float)(i*3+j+1)/10) +1; //內層迴圈第一次執行(i = 0)從0~2，分別為 0.1+1, 0.2+1, 0.3+1，內層迴圈第二次執行(i = 1)從 0~2，分別為0.4+1, 0.5+1, 0.6+1
```

7. 請問以下程式執行結果為何？

※假設 zippo 的起始記憶體位置為 0x7fff59dac460

```
#include<stdio.h>
int main()
{
        int zippo[4][2] = {{27,79}, {48,40}, {14,88}, {84,55}};

        printf("   zippo = %p,     zippo+1 = %p\n", zippo, zippo+1);
        printf("zippo[0] = %p, zippo[0]+1 = %p\n", zippo[0], zippo[0]+1);
        printf("  *zippo = %p,    *zippo+1 = %p\n",*zippo, *zippo+1);

        printf("zippo[3][1] = %d\n", zippo[3][1]);
        printf(" *zippo[2] = %d\n", *zippo[2]);
        printf("  **zippo = %d\n", **zippo);
        printf("   zippo[1][1] = %d\n", zippo[1][1]);
        printf("*(*(zippo+3) +1)  = %d\n", *(*(zippo+3) +1) );
}
```

7.

zippo = 0x7fff59dac460, zippo+1 = 0x7fff59dac468 //zippo以兩個int為一個單位，因此zippo+1的位址會比zippo多8個bytes(int佔4個bytes)

zippo[0] = 0x7fff59dac460, zippo[0]+1 = 0x7fff59dac464 //zippo[0]指向{27, 79}這個陣列，其中以一個int為單位，因此zippo[0]+1(等同於zippo[0][1])的位址會比zippo[0]多4個bytes

*zippo = 0x7fff59dac460, *zippo+1 = 0x7fff59dac464 //*zippo指向{27, 79}這個陣列，其中以一個int為單位，因此*zippo+1的位址會比*zippo多4個bytes

zippo[3][1] = 55

zippo[2] = 14 //zippo[0]指向{27, 79}，zippo[1]指向{48, 40}，zippo[2]指向{14, 88}，所以(zippo[2])的值為等同於zippo[2][0]等於14

**zippo = 27 //*zippo指向{27, 79}，所以*(*zippo)的值等同於zippo[0][0]等於27

zippo[1][1] = 40

((zippo+3) +1) = 55 //zippo 指向{27, 79}，zippo+3 指向{84, 55}，*(zippo+3)指向 84，*(zippo+3)+1 指向 55，*(*(zippo+3)+1)的值即為 55

8. 請問在以下的四種情況下，***ptr** 和 ***(ptr+2)** 分別表示什麼數值？

(a)

```
int *ptr;
int torf[3][2] = {63,73,42,79,80,26};
ptr = torf[0];
```

(b)

```
int *ptr;
int torf[3][2] = {10,54,77,27,49,8};
ptr = torf[1];
```

8.

a

//torf[0]指向{63 }

*(ptr) = 63

*(ptr+2) = 42

b

//torf[1]指向{77 }

*(ptr) = 77

*(ptr+2) = 49

9. 請問在以下的兩種情況下， **ptr 和 **(ptr+3)，**ptr+1，*(*ptr+2) 分別表示什麼數值？

(a)

```
int (*ptr)[2];
int torf[3][2] = {38,47,89,17,65,59};
ptr = torf;
```

9.

(a)

**ptr = 38 //等於*(ptr[0])，等於ptr[0][0]

**(ptr+3) = 0 //ptr指向{38, 47}，ptr+1指向{89, 17}，ptr+2指向{65, 59}，ptr+3超出範圍

ptr+1 = 39 //(ptr) + 1

*(*ptr+2) = 89 //*ptr 為 ptr[0][0](38 的位址，(*ptr) + 1 為 ptr[0][1](47 的位址)，(*ptr) + 2 為 ptr[1][0](89 的位址)，此處 ptr[1][0]為 ptr[0][1]的下一個位址

11. 參考以下每一行註解，寫出每一行對應的程式碼

```
int a=3, b;
int *p, *q;

___(a)___;  // p 指向 a
___(b)___;  //b 等於 p 指向的內容
___(c)___;  //q 指向 p 所指的內容
___(d)___;  //利用 q 將所指向的內容改成 4
```

11.

a. p=&a

b. b=*p

c. q=p

d. *q=4

4. 根據註解填寫相對應之程式碼

```
#include <stdio.h>
int main(void)
{

    int A[5][10];

    // 假設此區塊 對A 給值
        ...

    /* 宣告一個整數指標 p , 當 p+1 時 會指向下一個int。
       並指向A陣列的Row 0的啟始位置    */
    _____(a)_____;

    /* 宣告一個整數指標 q , 當 q+1 時 會指向A陣列的下一個ROW。*/
    _____(b)_____;

    /* 令 p 指向A的Row 2 */
    _____(c)_____;  ;

    /* 利用 p 將A[2][3]印出 (使用陣列模式) */
    printf("%d\n",_____(d)_____);

    /* 利用 p 將A[2][3]印出 (使用指標模式) */
    printf("%d\n",_____(e)_____);

    /* 令 q 指向 A的Row 2 */
    _____(f)_____;  ;

    /* 令 q 往下移一個row */
    _____(g)_____;  ;

    /* 利用 q 將A[3][3]印出 */
    printf("%d\n",_____(h)_____);


}
```

```c
#include <stdio.h>

int main(void)
{

    //int A[5][10];

    // 假設此區塊 對A 給值

    ...

    // 宣告一個整數指標 p , 當 p+1 時 會指向下一個int。
    //  並指向A陣列的Row 0的啟始位置
    int *p = A;  // 或 int *p = A[0]; 或 int *p = &A[0][0];


    // 宣告一個整數指標 q , 當 q+1 時 會指向A陣列的下一個ROW。
    int(*q)[10];

    // 令 p 指向A的Row 2
    p = A[2];

    // 利用 p 將A[2][3]印出 (使用陣列模式)
    printf("A[2][3] = %d\n", p[3]);

    // 利用 p 將A[2][3]印出 (使用指標模式)
    printf("A[2][3] = %d\n", *(p + 3));

    // 令 q 指向 A的Row 2
    q = &A[2];

    // 令 q 往下移一個row
    q++;

    // 利用 q 將A[3][3]印出
    printf("A[3][3] = %d\n", *(*q + 3));

}
```