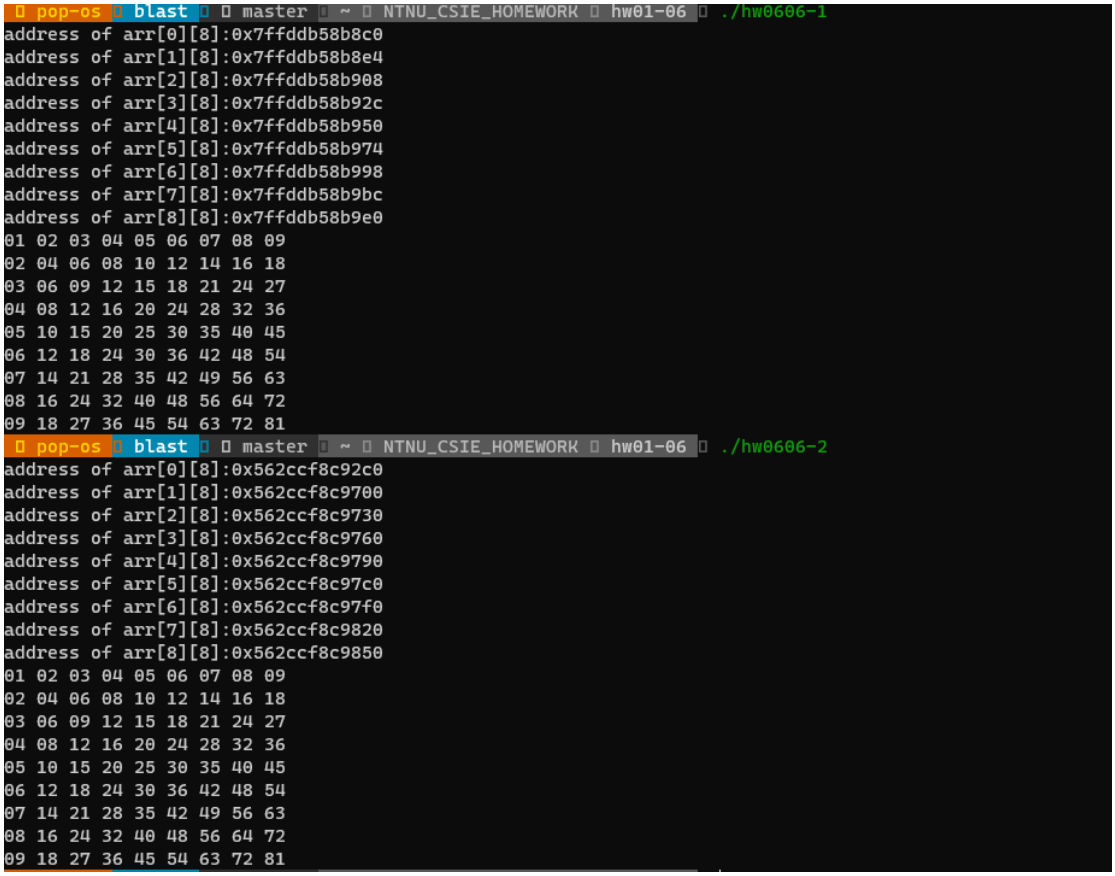
第一個使用陣列開出來，因此記憶體使用是連續且大小固定的。
第二個適用指標開出來的，因此記憶體使用是隨時增加直到跟第一個大小一樣且不連續的。
實驗:
分別把兩個程式的第 i 行第 8 個記憶體位置印出來
Hw0606-1.c(for array):

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

int main(){
    int32_t a[9][9] = {0};
    for( size_t i = 0 ; i < 9 ; i++ ){
        printf("address of arr[%lu][8]:%p\n",i,&a[i][8]);
        for( size_t j = 0 ; j < 9 ; j++ ){
            a[i][j] = ( i + 1 ) * ( j + 1 );
        }
    }
    for( size_t i = 0 ; i < 9 ; i++ ){
        for( size_t j = 0 ; j < 9 ; j++ ){
            printf( "%02d ", a[i][j] );
        }
        printf( "\n" );
    }
    return 0;
}
```

Hw0606-2.c(for malloc):

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

int main(){
    int32_t *a[9] = {0};
    for( size_t i = 0 ; i < 9 ; i++ ){
        a[i]=(int32_t*)malloc(9*sizeof(int32_t));
        printf("address of arr[%lu][8]:%p\n",i,&a[i][8]);
        for( size_t j = 0 ; j < 9 ; j++ ){
            a[i][j] = ( i + 1 ) * ( j + 1 );
        }
    }
    for( size_t i = 0 ; i < 9 ; i++ ){
        for( size_t j = 0 ; j < 9 ; j++ ){
            printf( "%02d ", a[i][j] );
        }
        printf( "\n" );
    }
    return 0;
}
```

得到以下結果



可以發現 hw0606-1 的第一行和第二行記憶體相差 e4-c0=(24)16=(36)10
剛好差九個 int 的大小，因此為連續的
Hw0606-2 的第一行跟第二行 700—2c0 很明顯差超過 36byte，因此可以得到為不連續的記憶體空間。
接著證明 hw0606-2 記憶體空間是逐行分配的，將原本的 malloc 移動到 printf 之後，如下圖:

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

int main(){
    int32_t *a[9] = {0};
    for( size_t i = 0 ; i < 9 ; i++ ){
        printf("address of arr[%lu][8]:%p\n",i,&a[i][8]);
        a[i]=(int32_t*)malloc(9*sizeof(int32_t));
        for( size_t j = 0 ; j < 9 ; j++ ){
            a[i][j] = ( i + 1 ) * ( j + 1 );
        }
    }
    for( size_t i = 0 ; i < 9 ; i++ ){
        for( size_t j = 0 ; j < 9 ; j++ ){
            printf( "%02d ", a[i][j] );
        }
        printf( "\n" );
    }
    return 0;
}
```

得到的結果

```
 pop-os  blast  master  ~  NTNU_CSIE_HOMEWORK  hw01-06  vim hw0606-2.c
 pop-os  blast  master  ~  NTNU_CSIE_HOMEWORK  hw01-06  ./hw0606-2
address of arr[0][8]:0x20
address of arr[1][8]:0x20
address of arr[2][8]:0x20
address of arr[3][8]:0x20
address of arr[4][8]:0x20
address of arr[5][8]:0x20
address of arr[6][8]:0x20
address of arr[7][8]:0x20
address of arr[8][8]:0x20
01 02 03 04 05 06 07 08 09
02 04 06 08 10 12 14 16 18
03 06 09 12 15 18 21 24 27
04 08 12 16 20 24 28 32 36
05 10 15 20 25 30 35 40 45
06 12 18 24 30 36 42 48 54
07 14 21 28 35 42 49 56 63
08 16 24 32 40 48 56 64 72
09 18 27 36 45 54 63 72 81
 pop-os  blast  master  ~  NTNU_CSIE_HOMEWORK  hw01-06
```

可以發現每一行的第八個記憶體位置都一樣，沒有被初始化到不同的位置，依此得到使用 malloc 寫法的記憶體是逐行分配的。