計算機概論 作業六之二

醫工一 葉芸茜 B812110004

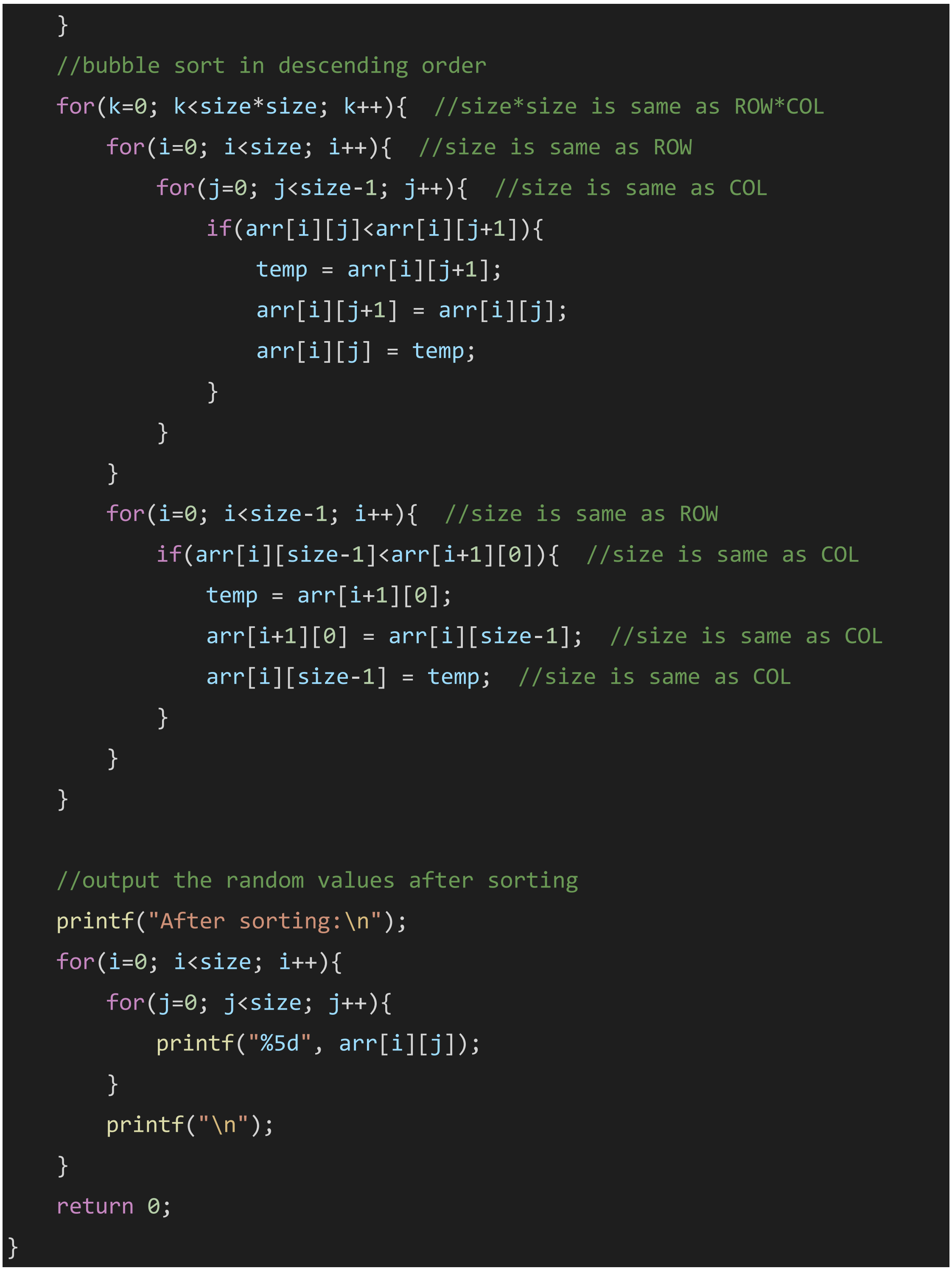
HW 6-2

一、version 1

（一）程式碼

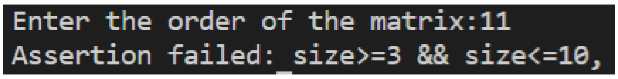
使用者輸入二維陣列大小，系統產生 10~100 的亂數，並分別印出未排序前及以二維陣列氣泡排序法排序後的結果

```
#include<stdio.h>
#include<assert.h>
#include<time.h>
#include<stdlib.h>
#define n 10
//讓使用者輸入陣列大小，由系統產生數值亂數，之後將二維陣列做氣泡降排序法後輸出

int main(){
    int arr[n][n], size, i, j, k, temp;
    //input the array size of the 2D array
    printf("Enter the order of the matrix:");
    scanf("%d", &size);
    //if the array size is smaller than 3 or larger than 10, main function
will stop
    assert(size>=3 && size<=10);
    //generate random values of the 2D array in the range of 10 to 100
    srand(time(NULL));
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            arr[i][j] = rand()%91 + 10;
        }
    }
    //output the random values before sorting
    printf("Before sorting:\n");
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            printf("%5d", arr[i][j]);
        }
        printf("\n");
```

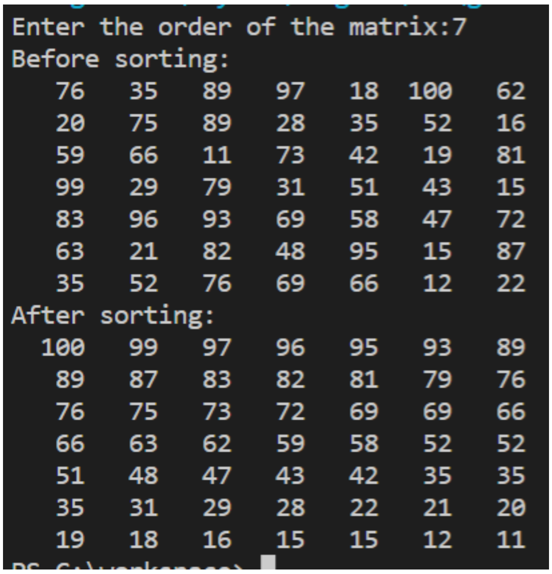


```
    }
    //bubble sort in descending order
    for(k=0; k<size*size; k++){  //size*size is same as ROW*COL
        for(i=0; i<size; i++){  //size is same as ROW
            for(j=0; j<size-1; j++){  //size is same as COL
                if(arr[i][j]<arr[i][j+1]){
                    temp = arr[i][j+1];
                    arr[i][j+1] = arr[i][j];
                    arr[i][j] = temp;
                }
            }
        }
        for(i=0; i<size-1; i++){  //size is same as ROW
            if(arr[i][size-1]<arr[i+1][0]){  //size is same as COL
                temp = arr[i+1][0];
                arr[i+1][0] = arr[i][size-1];  //size is same as COL
                arr[i][size-1] = temp;  //size is same as COL
            }
        }
    }

    //output the random values after sorting
    printf("After sorting:\n");
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            printf("%5d", arr[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

（二）輸出結果



```
Enter the order of the matrix:11
Assertion failed: size>=3 && size<=10,
```



```
Enter the order of the matrix:7
Before sorting:
   76    35    89    97    18   100    62
   20    75    89    28    35    52    16
   59    66    11    73    42    19    81
   99    29    79    31    51    43    15
   83    96    93    69    58    47    72
   63    21    82    48    95    15    87
   35    52    76    69    66    12    22
After sorting:
  100    99    97    96    95    93    89
   89    87    83    82    81    79    76
   76    75    73    72    69    69    66
   66    63    62    59    58    52    52
   51    48    47    43    42    35    35
   35    31    29    28    22    21    20
   19    18    16    15    15    12    11
PS C:\workspace>
```

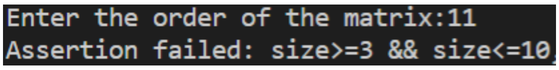
二、version 2

（一）程式碼

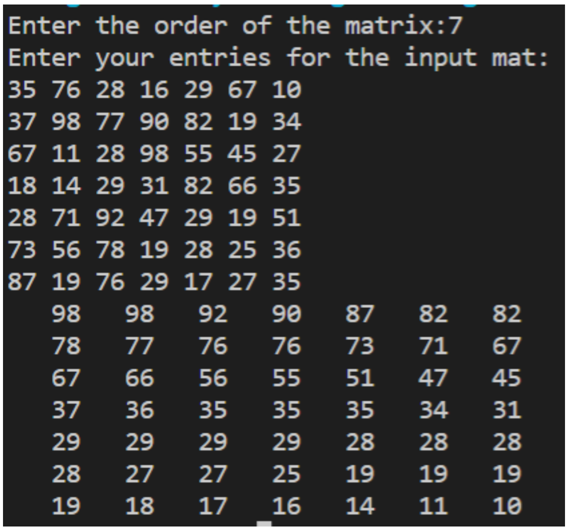
使用者輸入二維陣列大小及陣列內數值，以二維氣泡排序法(與 version 1 的排序法不同)排序後輸出

```
#include<stdio.h>
#include<assert.h>
#define n 10
//讓使用者自行輸入陣列大小及數值，之後將二維陣列做氣泡降排序法後輸出

int main(){
    int arr[n][n], size, i, j, k, r, temp;
    printf("Enter the order of the matrix:"); //輸入二維陣列大小
    scanf("%d", &size);
    assert(size>=3 && size<=10);
    printf("Enter your entries for the input mat:\n"); //輸入二維陣列數值
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            scanf("%d", &arr[i][j]);
        }
    }
    //bubble sort of 2d array in descending order
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            temp = arr[i][j];
            r = j+1;
            for(k=i; k<size; k++){
                while(r<size){
                    if(temp<arr[k][r]){
                        temp = arr[k][r];
                        arr[k][r] = arr[i][j];
                        arr[i][j] = temp;
                    }
                    r++;
                }
                r=0;
            }
        }
    }
    //using nested loops to output the 2d array
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            printf("%5d", arr[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

（二）輸出結果



```
Enter the order of the matrix:11
Assertion failed: size>=3 && size<=10
```



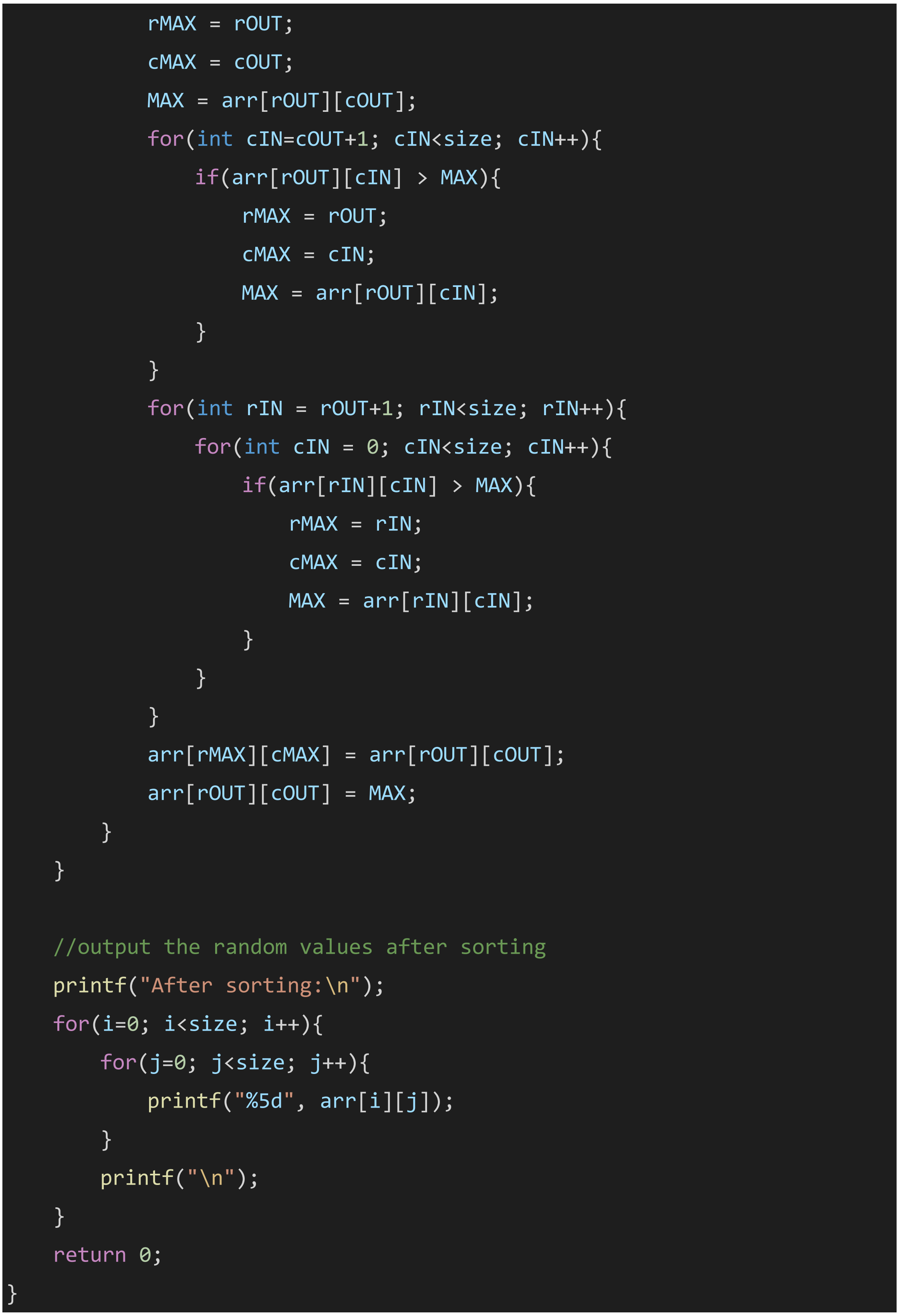
```
Enter the order of the matrix:7
Enter your entries for the input mat:
35 76 28 16 29 67 10
37 98 77 90 82 19 34
67 11 28 98 55 45 27
18 14 29 31 82 66 35
28 71 92 47 29 19 51
73 56 78 19 28 25 36
87 19 76 29 17 27 35
   98   98   92   90   87   82   82
   78   77   76   76   73   71   67
   67   66   56   55   51   47   45
   37   36   35   35   35   34   31
   29   29   29   29   28   28   28
   28   27   27   25   19   19   19
   19   18   17   16   14   11   10
```

三、version 3

（一）程式碼

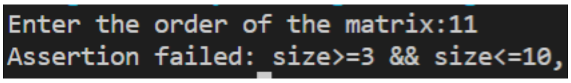
使用者輸入二維陣列大小，系統產生 10~100 的亂數，並分別輸出排序前及以選擇排序法排序後的結果

```
#include<stdio.h>
#include<assert.h>
#include<time.h>
#include<stdlib.h>
#define n 10
//讓使用者輸入陣列大小，由系統產生數值亂數，之後將二維陣列做選擇降排序法後輸出

int main(){
    int arr[n][n], size, i, j;
    //input the array size of the 2D array
    printf("Enter the order of the matrix:");
    scanf("%d", &size);
    //if the array size is smaller than 3 or larger than 10, main function
will stop
    assert(size>=3 && size<=10);
    //generate random values of the 2D array in the range of 10 to 100
    srand(time(NULL));
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            arr[i][j] = rand()%91 + 10;
        }
    }
    //output the random values before sorting
    printf("Before sorting:\n");
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            printf("%5d", arr[i][j]);
        }
        printf("\n");
    }
    //selection sort in descending order
    int rMAX, cMAX, MAX;
    for(int rOUT=0; rOUT<size; rOUT++){
        for(int cOUT=0; cOUT<size; cOUT++){
```

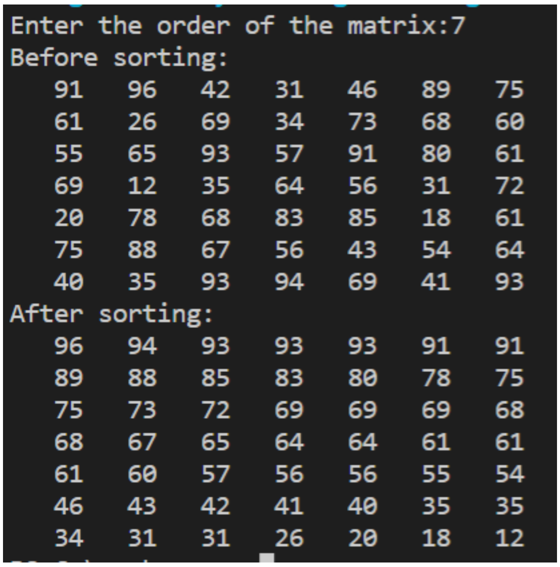


```
            rMAX = rOUT;
            cMAX = cOUT;
            MAX = arr[rOUT][cOUT];
            for(int cIN=cOUT+1; cIN<size; cIN++){
                if(arr[rOUT][cIN] > MAX){
                    rMAX = rOUT;
                    cMAX = cIN;
                    MAX = arr[rOUT][cIN];
                }
            }
            for(int rIN = rOUT+1; rIN<size; rIN++){
                for(int cIN = 0; cIN<size; cIN++){
                    if(arr[rIN][cIN] > MAX){
                        rMAX = rIN;
                        cMAX = cIN;
                        MAX = arr[rIN][cIN];
                    }
                }
            }
            arr[rMAX][cMAX] = arr[rOUT][cOUT];
            arr[rOUT][cOUT] = MAX;
        }
    }

    //output the random values after sorting
    printf("After sorting:\n");
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            printf("%5d", arr[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

（二）輸出結果



```
Enter the order of the matrix:11
Assertion failed: size>=3 && size<=10,
```



```
Enter the order of the matrix:7
Before sorting:
   91   96   42   31   46   89   75
   61   26   69   34   73   68   60
   55   65   93   57   91   80   61
   69   12   35   64   56   31   72
   20   78   68   83   85   18   61
   75   88   67   56   43   54   64
   40   35   93   94   69   41   93
After sorting:
   96   94   93   93   93   91   91
   89   88   85   83   80   78   75
   75   73   72   69   69   69   68
   68   67   65   64   64   61   61
   61   60   57   56   56   55   54
   46   43   42   41   40   35   35
   34   31   31   26   20   18   12
```

四、version 4

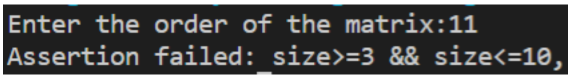
（一）程式碼

使用者輸入二維陣列大小，系統產生 10~100 的亂數，輸出未排序前的二維陣列。之後將二維陣列轉為一維陣列的形式進行氣泡排序法排序，並再次將一維轉為二維陣列輸出

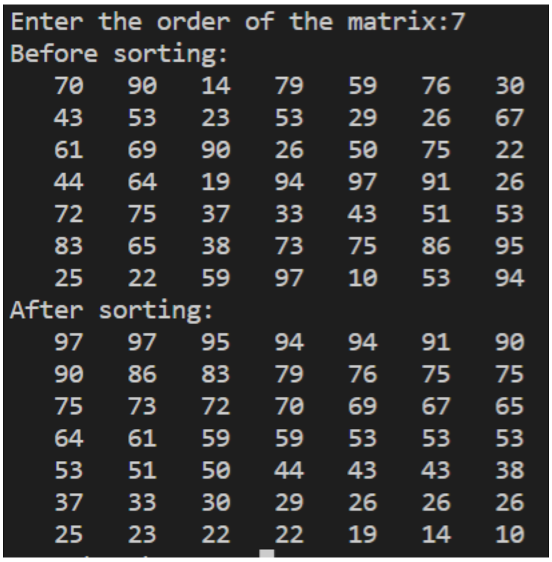
```
#include<stdio.h>
#include<assert.h>
#include<stdlib.h>
#include<time.h>
#define n 10
//這是一個將二維轉成一維陣列，再用氣泡排序法由大到小做排序，之後再轉成二維陣列輸出的麻煩過程
int bubble_sort(int a[], int num);
int main(){
   int arr[n][n], size, i, j, k=0, w=0;
   printf("Enter the order of the matrix:"); //input the arraysize of 2d array
   scanf("%d", &size);
   assert(size>=3 && size<=10);
   //generate random values of the 2D array in the range of 10 to 100
    srand(time(NULL));
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            arr[i][j] = rand()%91 + 10;
        }
    }
    //output the random values before sorting
    printf("Before sorting:\n");
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            printf("%5d", arr[i][j]);
        }
        printf("\n");
    }
   int arr2[size*size]; //declare a 1D array

   //2D array convert to 1D array
   for(i=0; i<size; i++){
     for(j=0; j<size; j++){
```

```c
         arr2[k] = arr[i][j];
         k++;
      }
   }
   //bubble sort of 1D array
   bubble_sort(arr2, size*size);

   //1D array convert to 2D array
   for(i=0; i<size; i++){
      for(j=0; j<size; j++){
         arr[i][j]=arr2[w];
         w++;
      }
   }
   //print 2D array
   printf("After sorting:\n");
   for(i=0; i<size; i++){
      for(j=0; j<size; j++){
         printf("%5d", arr[i][j]);
      }
      printf("\n");
   }
   return 0;
}

int bubble_sort(int a[], int num){
    int temp;
    for(int i=0; i<num-1; ++i){
        for(int j=0; j<num-1-i; ++j){
            if(a[j]<a[j+1]){
                temp = a[j+1];
                a[j+1] = a[j];
                a[j] = temp;
            }
        }
    }
   return 0;
}
```

（二）輸出結果



```
Enter the order of the matrix:11
Assertion failed: size>=3 && size<=10,
```



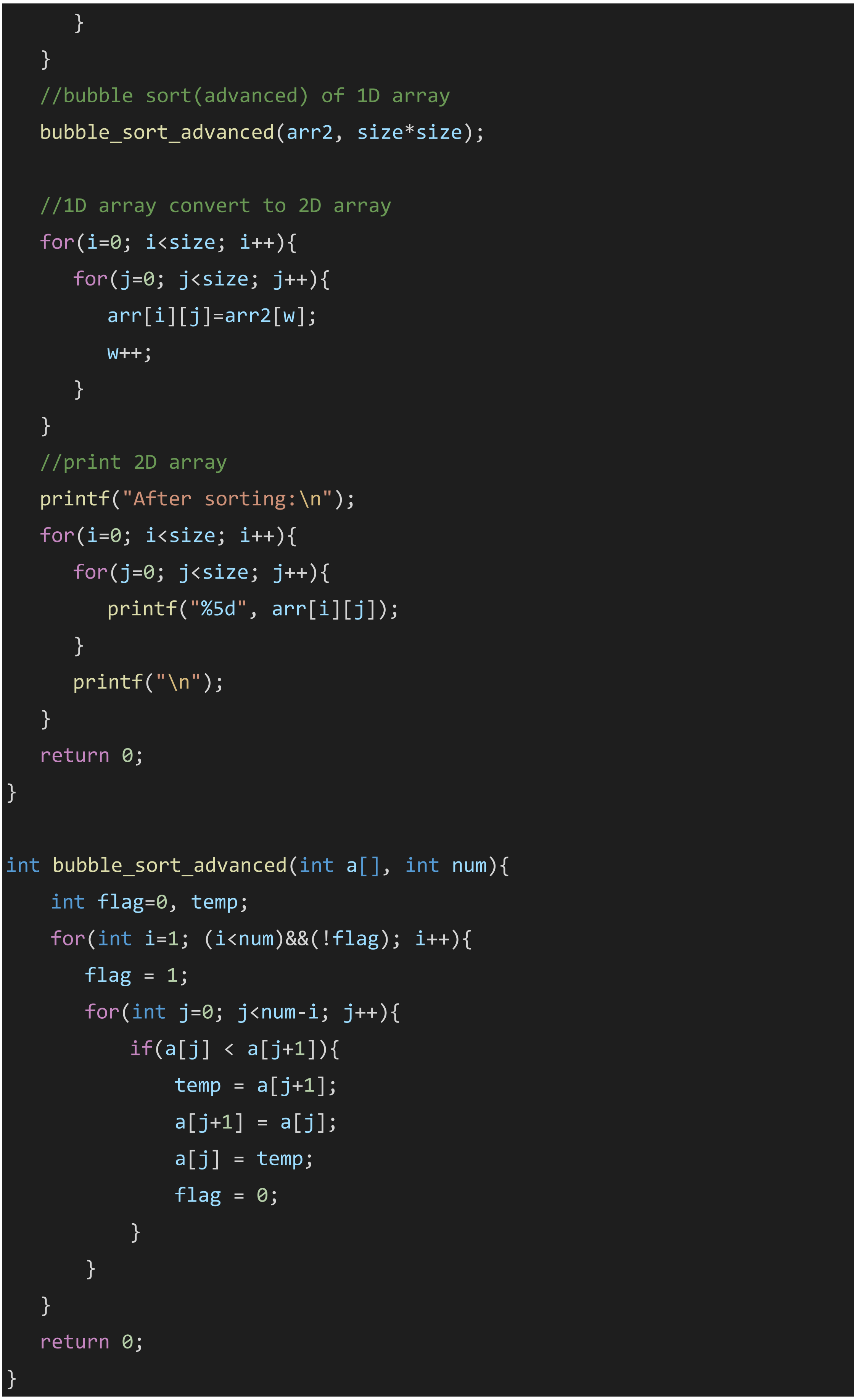
```
Enter the order of the matrix:7
Before sorting:
   70    90    14    79    59    76    30
   43    53    23    53    29    26    67
   61    69    90    26    50    75    22
   44    64    19    94    97    91    26
   72    75    37    33    43    51    53
   83    65    38    73    75    86    95
   25    22    59    97    10    53    94
After sorting:
   97    97    95    94    94    91    90
   90    86    83    79    76    75    75
   75    73    72    70    69    67    65
   64    61    59    59    53    53    53
   53    51    50    44    43    43    38
   37    33    30    29    26    26    26
   25    23    22    22    19    14    10
```

五、version 5

（一）程式碼

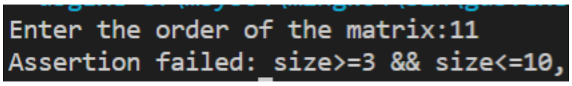
與 version 4 大致相同，但改以一維陣列氣泡排序法改良版進行排序

```
#include<stdio.h>
#include<assert.h>
#include<stdlib.h>
#include<time.h>
#define n 10
//這是一個將二維轉成一維陣列，再用氣泡排序法進階版由大到小做排序，之後再轉成二維陣列輸出的麻煩過程
int bubble_sort_advanced(int a[], int num);
int main(){
   int arr[n][n], size, i, j, k=0, w=0;
   printf("Enter the order of the matrix:"); //input the arraysize of 2d array
   scanf("%d", &size);
   assert(size>=3 && size<=10);
   //generate random values of the 2D array in the range of 10 to 100
    srand(time(NULL));
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            arr[i][j] = rand()%91 + 10;
        }
    }
    //output the random values before sorting
    printf("Before sorting:\n");
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            printf("%5d", arr[i][j]);
        }
        printf("\n");
    }
   int arr2[size*size]; //declare a 1D array

   //2D array convert to 1D array
   for(i=0; i<size; i++){
      for(j=0; j<size; j++){
         arr2[k] = arr[i][j];
         k++;
```

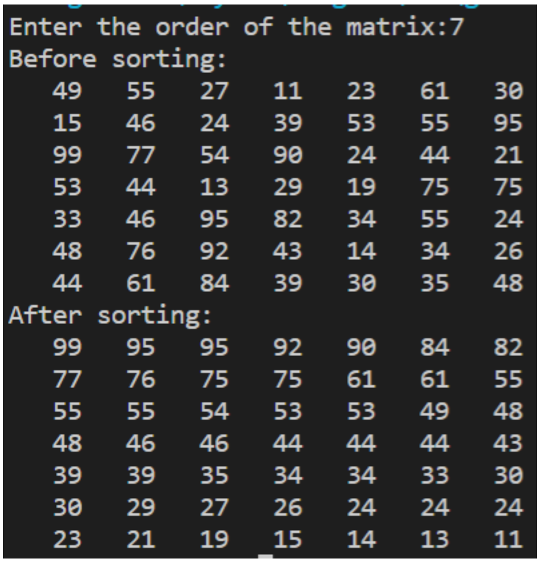


```
      }
   }
   //bubble sort(advanced) of 1D array
   bubble_sort_advanced(arr2, size*size);

   //1D array convert to 2D array
   for(i=0; i<size; i++){
      for(j=0; j<size; j++){
         arr[i][j]=arr2[w];
         w++;
      }
   }
   //print 2D array
   printf("After sorting:\n");
   for(i=0; i<size; i++){
      for(j=0; j<size; j++){
         printf("%5d", arr[i][j]);
      }
      printf("\n");
   }
   return 0;
}

int bubble_sort_advanced(int a[], int num){
    int flag=0, temp;
    for(int i=1; (i<num)&&(!flag); i++){
       flag = 1;
       for(int j=0; j<num-i; j++){
           if(a[j] < a[j+1]){
               temp = a[j+1];
               a[j+1] = a[j];
               a[j] = temp;
               flag = 0;
           }
       }
   }
   return 0;
}
```

（二）輸出結果



```
Enter the order of the matrix:11
Assertion failed: size>=3 && size<=10,
```



```
Enter the order of the matrix:7
Before sorting:
   49    55    27    11    23    61    30
   15    46    24    39    53    55    95
   99    77    54    90    24    44    21
   53    44    13    29    19    75    75
   33    46    95    82    34    55    24
   48    76    92    43    14    34    26
   44    61    84    39    30    35    48
After sorting:
   99    95    95    92    90    84    82
   77    76    75    75    61    61    55
   55    55    54    53    53    49    48
   48    46    46    44    44    44    43
   39    39    35    34    34    33    30
   30    29    27    26    24    24    24
   23    21    19    15    14    13    11
```

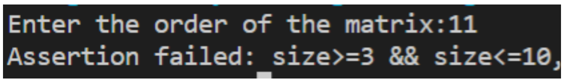
六、version 6

（一）程式碼

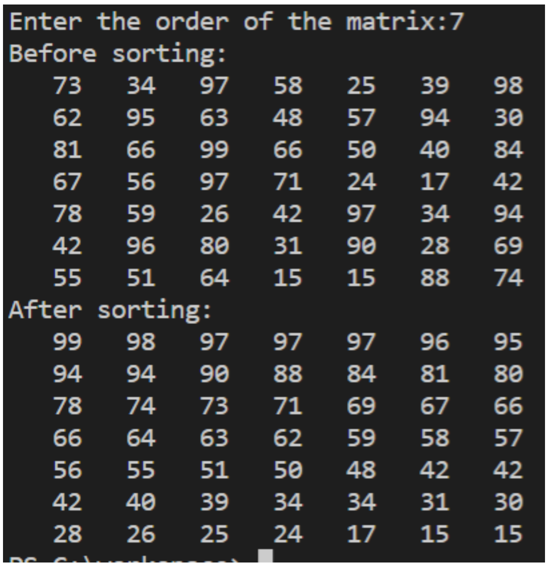
與 version 4 and 5 大致相同，但改以一維陣列選擇排序法進行排序

```
#include<stdio.h>
#include<assert.h>
#include<stdlib.h>
#include<time.h>
#define n 10
//這是一個將二維轉成一維陣列，再用選擇排序法做降排序，之後再轉成二維陣列輸出的麻煩過程
int selection_sort(int a[], int num);
int main(){
   int arr[n][n], size, i, j, k=0, w=0;
   printf("Enter the order of the matrix:"); //input the arraysize of 2d array
   scanf("%d", &size);
   assert(size>=3 && size<=10);
   //generate random values of the 2D array in the range of 10 to 100
    srand(time(NULL));
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            arr[i][j] = rand()%91 + 10;
        }
    }
    //output the random values before sorting
    printf("Before sorting:\n");
    for(i=0; i<size; i++){
        for(j=0; j<size; j++){
            printf("%5d", arr[i][j]);
        }
        printf("\n");
    }
   int arr2[size*size]; //declare a 1D array
   //2D array convert to 1D array
   for(i=0; i<size; i++){
      for(j=0; j<size; j++){
         arr2[k] = arr[i][j];
         k++;
      }
   }
   //selection sort of 1D array
   selection_sort(arr2, size*size);

   //1D array convert to 2D array
   for(i=0; i<size; i++){
      for(j=0; j<size; j++){
         arr[i][j]=arr2[w];
         w++;
      }
   }
   //print 2D array
   printf("After sorting:\n");
   for(i=0; i<size; i++){
      for(j=0; j<size; j++){
         printf("%5d", arr[i][j]);
      }
      printf("\n");
   }
   return 0;
}

int selection_sort(int a[], int num){
    int temp;
    for(int i=0; i<num-1; i++){
        int min_idx = i;
        for(int j=i+1; j<num; j++){
            if(a[j] > a[min_idx]){
                min_idx = j;
            }
        }
        temp = a[i];
        a[i] = a[min_idx];
        a[min_idx] = temp;
   }
   return 0;
}
```

（二）輸出結果



```
Enter the order of the matrix:11
Assertion failed: size>=3 && size<=10,
```



```
Enter the order of the matrix:7
Before sorting:
   73   34   97   58   25   39   98
   62   95   63   48   57   94   30
   81   66   99   66   50   40   84
   67   56   97   71   24   17   42
   78   59   26   42   97   34   94
   42   96   80   31   90   28   69
   55   51   64   15   15   88   74
After sorting:
   99   98   97   97   97   96   95
   94   94   90   88   84   81   80
   78   74   73   71   69   67   66
   66   64   63   62   59   58   57
   56   55   51   50   48   42   42
   42   40   39   34   34   31   30
   28   26   25   24   17   15   15
```

- 討論

因為陣列的宣告必須先給以陣列大小，但因為陣列的大小是設定給以使用者輸入，因此不能以變數做宣告，如下為錯誤示範:

```
#include<stdio.h>
int main(void){
    int n, arr[n][n];
    printf( “please enter the array size” );
    scanf( “%d” , &n);
}
```

所以我先 define n 為 10(當 array size 超過 10 則會 assertion failed)，給予一個值並確保使用者輸入的值不會超過其大小。

好像還有一種方法是利用動態記憶體配置，使用 malloc 函數來配置記憶體空間，之後由 free 函數來釋放之前 malloc 函數的記憶體空間，所使用的 header file 為<stdlib.h>，但因為好像會牽扯到 pointer 的概念，所以沒使用。