1.(10%) What will be the value of Counter after the execution of the following program?

```
main ()
    {
        int Counter = 0;
        float Total = 0.00;

        while (Total != 1.00)
          {
            Total = Total + 0.01;
            Counter = Counter + 1;
          }
    }
```

2.(15%) The procedure for executing a "Ripple Sort" is outlined as below:

(a) Start with the first entry.
(b) Look at the remaining entries one by one. Whenever you find a smaller entry, swap it with the first entry, so that the first entry always holds the candidate for the smallest entry.
(c) Shrink how you look at the list by starting with the second entry and look at the remaining entries (3, 4, and so on).
(d) Continue step (c) until all items are worked through.

Write a program in C language to implement the above procedure.

3.(10%) Check which of the following are usually performed by a program loader:

(A) Allocate space for the program and data.
(B) Determine the print queue length.
(C) Relocate the code or set a base register.
(D) Check the program for system calls.
(E) Initialize the page table.

4.(10%) When executing $\sum_{i=1}^{k+1} a_{n-i} x^{n-i}$, at least __(A)__ times of addition and __(B)__ times of multiplication will be performed.

5.(5%) Explain the differences between a round-off error and a truncation error.

6.(10%) 試申述電腦在工業工程的應用。

7.(10%) 今有一個實數陣列內存有數值:
double a[100];
請用一個 for 迴圈 (for loop) 去找出平均值、最大值及最小值。

8.(10%) 一個二元聯立方程式問題如下：

x + y = 2.2
x - 2y = -1.2

今有一數學軟体可求出 x, y 的值，並分別存於 a, b 兩個 double 變數中，他們原始的宣告爲：
double a, b;

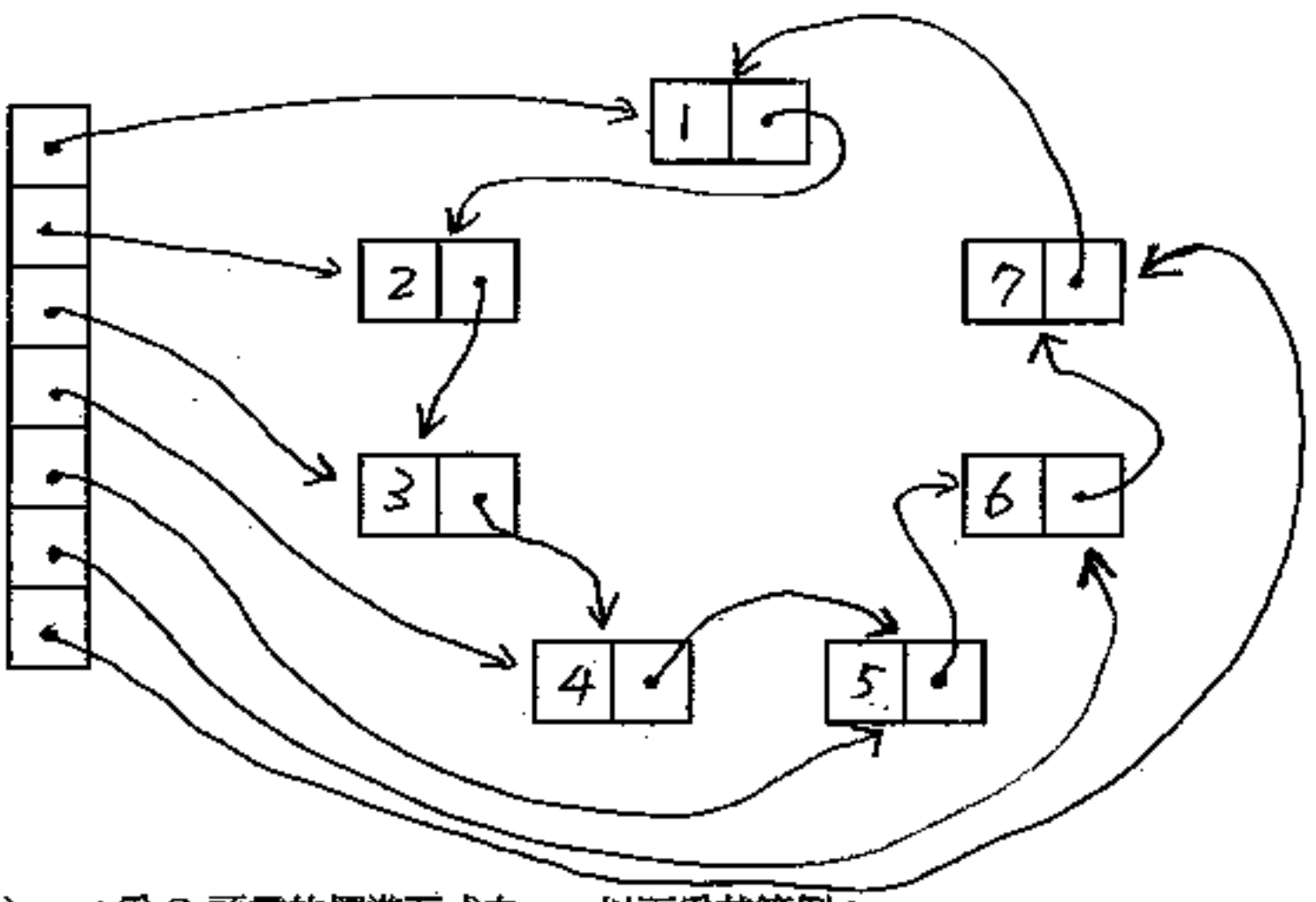
請寫一段 C 程式去驗証該軟体的答案是否正確

9.(10%) 定義一個 node 的結構体:

```
struct node
      {
      int num;
      stuct node *next;
      }
```

寫一段程式去建構一個由七個 node 所形成的迴圈及由七個 node 的地址(address) 所組成的陣列(Array):

圖例：



10.(10%) qsort 爲 C 語言的標準函式之一，以下爲其範例：

What is qsort? This function is C's standard sorting function, the quicksort routine described in Application Program 8.3. It has the prototype:

```
void qsort (void *buf, int num, int size, int (*compare) (const void *, const void *));
```

where compare is a pointer to a function, buf is a pointer to the array being sorted, num is the number of elements in the array, and size is the number of bytes of a single element.

By using a pointer to the function compare, qsort is able to call a function that we define. The function pointed to by compare (that is; the function we define) must meet the following requirements:

1. It must have two arguments, both of type const void *.
2. It must return an integer.
3. The return integer must be dependent upon the relative magnitudes of the two arguments. To sort a list in increasing order, the return integer must be
   greater than 0 if the first argument is greater than the second
   less than 0 if the first argument is less than the second
   equal to 0 if the first and second arguments are equal

The short program that follows uses qsort to sort the array list[6]. We have defined the function sortinc, which conforms to the three requirements just given.

The prototype for qsort is contained in stdlib.h.

The function sortinc has two arguments of type const void *.

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

int sortinc(const void *aa, const void *bb);

void main(void)
{
        int i;
        double list[6]={1.3, 8.7, 9.3, 8.5, -3.4, 4.1};

        qsort(list, 6, sizeof(double), sortinc);
        for(i=0; i<6; i++) printf("%f\n", list[i]);
}

int sortinc(const void *aa, const void *bb)
{
        double cc, dd;
        int ee;

        cc=*(double *)aa;
        dd=*(double *)bb;

        if (cc>dd) ee=1;
        if (cc<dd) ee=-1;
        if (cc==dd) ee=0;

        return (ee);
}
```

The array to be sorted is list[6].

Each element of list[ ] is a double.

We pass the address of sortinc to qsort.

Here, sortinc returns an int.

Printing the sorted list.

We convert aa and bb to be type double*.

The unary * operator acting on type double* points to a double. This matches the types of cc and dd.

Creating the return integer that meets the requirements for sorting a list in increasing order.

今二度空間的一個點(x, y) ，用 C 的 structure 表示:

```
struct point
    {
    float x;
    float y;
    };
```

又今有一個 100 個點(100 個 struct point)的陣列(Array):

```
stuct point[100];
```

試寫一段程式利用 qsort 函式將這 100 個點依其離原點的距離由小到大排列。