

113年公務人員特種考試警察人員、一般警察人員、 國家安全局國家安全情報人員及移民行政人員考試試題

考試別：國家安全情報人員考試

等別：三等考試

類科組別：資訊組（選試英文）

科目：計算機概論

考試時間：2小時

座號：_____

※注意：(一)禁止使用電子計算器。

(二)不必抄題，作答時請將試題題號及答案依照順序寫在試卷上，於本試題上作答者，不予計分。

(三)本科目除專門名詞或數理公式外，應使用本國文字作答。

一、請將以下 IEEE-754 單倍精確度 (32 bit) 浮點數格式 (二進位)，轉成「十進位浮點小數」。

Sign (1 bit)	Exponent (8 bits, bias=127)	Fraction (23 bits)
0	10000011	10100000000000000000000

再請針對十進位浮點小數 -17.125 表達成 IEEE-754 格式後，寫出其 Exponent 部位「最後 4 bits 的二進位值」。最後請將"十六進位資料 C0F80000"，表示成 IEEE-754 單倍精確度浮點數格式，寫出 Fraction 部位「最前面 4 bits 的二進位值」，和此"十六進位資料"的「十進位浮點小數」。(25分)

二、臺灣科技公司規劃網路，其中財務部門使用 C 等級 (class C) 的網路系統並手動設定 IP，子網路遮罩為 255.255.255.224。其中，財務主管小明的 IP 為 140.123.100.89，員工佳佳的 IP 為 140.123.100.182。請寫出小明的網路位址、廣播位址 (以十進位表示)；以及說明財務部門的網路可以切割成幾個子網路、每個子網路設定可用主機電腦數量。最後，請說明手動設定電腦 IP 位址時，子網路遮罩 (Subnet Mask) 與預設閘道的功能。(25分)

三、某城市有景點 A~E，其連接的道路均為單行道，單行道起點、終點與距離的資訊為：A->B (20)、A->C (25)、A->D (11)、B->C (25)、B->E (18)、C->D (27)、C->E (12)、D->B (24)、E->A (22)。請畫出"起始點為 B"之廣度優先走訪樹狀圖，其中每一個景點只出現一次。另外請畫出根節點"起始點為 B"，其他景點為葉節點的展開樹，找出 B->D 景點的所有路徑與最短路徑，以及所經過之景點及其距離。最後，請說明上面最短路徑走訪方法的演算步驟。(25分)

四、請完成計算中位數與眾數兩個方法(I)~(V)空格程式，使程式正常運作，以輸出 Mean=1, Mode=5。其中，中位數是將一組數值資料由小到大排列後，取最中間的數值為中位數。若資料個數為奇數，則取最中間的數值為中位數，例如 1, 2, 3, 3, 4, 6, 7, 7, 9 的中位數是 4；若資料個數為偶數，則取最中間兩個數值的平均為中位數，例如 1, 2, 3, 3, 4, 6, 7, 7, 8, 9 的中位數是 $(4+6)/2=5$ 。眾數，則是指一組數字中出現次數最頻繁的數字。例如，2、3、3、5、7、10 的眾數是 3。最後請說明此程式是否有不安全的地方，以及其原因。(25 分)

<pre>#include <iostream> #include <algorithm> using namespace std; class Compute { public: void setData(int *, int); double getMedian(); double getMode(); private: int *data, d_size; // 取得 key 在 map 的 中的索引 int getIndex(int d[2][20], int key, int c) { for (int i=0; i<c; i++) if (key==d[0][i]) return i; return -1; } //取得最大頻率值的 key int getMaxValue(int d[2][20], int c) { int maxKey = d[0][0]; int maxValue = d[1][0]; for (int i=1; i<c; i++) if (maxValue<d[1][i]) { maxKey = d[0][i]; maxValue = d[1][i]; } return maxKey; } }; //設定資料並完成排序 void Compute::setData(int *d, int s) { data = d; d_size = s; sort(data, data+d_size); }</pre>	<pre>// 求中位數，若數量為奇數，回傳中間值，否則 // 回傳中間平均值 double Compute::getMedian() { if (d_size%2== <u>(I)</u>) // (I) return data[<u>(II)</u>]; // (II) else return (data[(d_size/2)-1]+ <u>(III)</u>)/2.0; //(III) } //求眾數，統計數列各值出現頻率，找最大 double Compute::getMode() { //d_map[0]紀錄 Key, d_map[1]紀錄頻率值 int d_map[2][20]={0}; //d_map 數量 int map_count=0; for (int i=0; i<d_size; i++) { int index = getIndex(d_map, data[i], map_count); if (index== <u>(IV)</u>) { // (IV) d_map[0][map_count]=data[i]; d_map[1][map_count++]=1; } else d_map[1][<u>(V)</u>]=d_map[1][index]+1; //(V) } return getMaxValue(d_map, map_count); } int main() { int data[10] = {5, 3, -4, 3, -1, 5, -1, -3, 5, -2}; Compute p; p.setData(data, 10); cout<<"Mean="<<p.getMedian()<<" "; cout<<"Mode="<<p.getMode()<<endl; }</pre>
--	---