ME2008_C程式語言 HW＃5　LED點矩陣中文顯示

Date: 2013/3/30 posted.
TA: Tsai, Call Ext. 7261

OCR

please click this to see OCR-Version of HW5-Notes

Part1 – 11位元浮點數

完成 (1 , 5 , 5)11位元浮點數的心得和(表格答案)。表格請參考於3/28 公告之pdf檔案，自行製作 (1, 5, 5)表格，因表格數目過於龐大，故於四個角落各取12個數值共48個數值，表示其規則性即可。

Part2 – LED點矩陣中文顯示

參考HWbitmapINT.cpp的例子，完成以下小題。

1. 中文姓名直式列印。
2. 數字(學號)三碼橫式列印。

P.S.--

1. 姓名，每一字以**16 X 16** 規格列印。
2. 數字(本程式碼)以8 X 8方式列印。
3. 以正式的C# / C，**cin / cout** 和 **printf / scanf**方式各製作函式版本的列印程式碼。

HW5作業告知：

1. W07_(4月2日)前完成，你中文姓名的 " 繪圖"，並完成本例的compiler可產生 .exe 的動作。
2. HW5 正式作業報告，請在 4 / 9日中午前完成上繳BB動作，將做 計分 的評量。(上傳part1、part2 完成之輸出與程式碼)

有任何問題請至E1 – 411 ，謝謝。

參考

1. RENOIR website : [Homework]-[HW5] CLick this for HW5
2. C source - HWbitmapINT.cpp (Here)
3. NTUST BB Blackboard System CLick for BB System

4. NTUST site
5. You should Check out this:PDF file about IEEE754

C Source

```
// Name: B10003571   --HW5_Demo
// Date: Mar-2013    --
// GOAL:
// NOTE:  This VECTOR CPP program CAN be complied/run in Microsoft (VCPP-v9.0)
Environment.
//  But IS NOT ABLE to pass the SYNTAX checking in CHIDE Enviroment.
//
#include <iostream>
// #include <vector>

using namespace std ;

// 點矩陣文字
int  main() {

    int  i , j  , n  ;
    cout << "> 輸入中文字所對應的 8 個點矩數字 : " ;

    // vector<int>    no(8) ;
    // vector<bool>   bitmap(8) ;
    int no[8];
    int bitmap[8];


    // 儲存文字點矩陣資料
    for ( i = 0 ; i < 8 ; ++i ) cin >> no[i] ;

    cout << endl ;
    for (i=7; i>= 0;  i--){
         cout << i <<( 1 == 1 ? "*" : "  " ) ;
    }
    cout <<endl;
    // 迴圈重複每一行
    for ( i = 0 ; i < 8 ; ++i ) {

        n = no[i] ;

        // 將每一列的各個格子值存入 bitmap 矩陣中
        for ( j = 7 ; j >= 0 ; --j ) {
            bitmap[j] = n % 2 ;
            n /= 2 ;
        }

        // 列印每一列
        for ( j = 0 ; j < 8 ; ++j )
            cout << ( bitmap[j] ? " *" : "  " ) ;

        cout << endl ;

    }

    return 0 ;

}
```

Date: 4/2/2013 12:09:26 PM

Program Output Snapshots

1. Sample -8By8

```cpp
using namespace std ;

// 點矩陣文字
int  main() {

    int  i , j  , n  ;
    cout << "> 輸入中文字所對應的 8 個點矩數字 (Each 0~255): "

    // vector<int>    no(8) ;
    // vector<bool>   bitmap(8) ;
    int*  no;
    int bitmap[8];
    no = new int [8];

    // 儲存文字點矩陣資料
    for ( i = 0 ; i < 8 ; ++i ) cin >> no[i] ;

    cout << endl ;
    for (i=7; i>= 0;  i--){
        cout << i <<( 1 == 1 ? "*" : "  " ) ;
    }
    cout <<endl;
    // 迴圈重複每一行
    for ( i = 0 ; i < 8 ; ++i ) {

        n = no[i] ;

        // 將每一列的各個格子值存入 bitmap 矩陣中
        for ( j = 7 ; j >= 0 ; --j ) {
            bitmap[j] = n % 2 ;
            n /= 2 ;
        }

        // 列印每一列
        for ( j = 0 ; j < 8 ; ++j )
            cout << ( bitmap[j] ? " *" : "  " ) ;

        cout << endl ;

    }

    return 0 ;

}
```

Note:

Dynamic Array --

1) Declare a POINTER :: int* nptr;

2) Request to allocate spaces :: nptr = new int [Num]

```
Ch Ch-V70
E:/EElab> cl /EHsc HWbitDynamic.cpp
Microsoft (R) 32-bit C/C++ Optimizing Compiler Versio
Copyright (C) Microsoft Corporation.  All rights rese

HWbitDynamic.cpp
Microsoft (R) Incremental Linker Version 10.00.30319.
Copyright (C) Microsoft Corporation.  All rights rese

/out:HWbitDynamic.exe
HWbitDynamic.obj
E:/EElab> HWbitDynamic.exe < mybitdata.txt
> 輸入中文字所對應的 8 個點矩數字 (Each 0~255):
7*6*5*4*3*2*1*0*
        * * * *
 * * * * * * *   *
     *         *
     *         *
 * * * * * * * *
 * * * * * * *   *
   * * * * * * * *
 *
E:/EElab>
```

2. Toy Precsion :7Bit

Q1: which 48 numbers ? (Exact Number =? , Ex: 2, 4, 8, ...)

Q2: 12.5, 12.55, 13.75, -14.3, -11.275, 10.8725 - How to Express them ? (Why/ Why Not ?)

Toy precision, using IEEE rules

7 bits (only 2^7=128 different possibilities)

- 1 sign bit, S, (0 means positive, 1 means negative)
- 3 bits for the exponent, E (000,001,010,011,100,101,110,111)
- 3 bits for the mantissa, M

Rules

- If $1 \le E \le 6$, then value is $-1^S \times 2^{E-3} \times 1.M$
  - 2^{E-3} takes on values 0.25, 0.5, 1, 2, 4, 8
  - 1.M values: 1, 1.125, 1.25, 1.375, 1.5, 1.6125, 1.75, 1.8725
  - Minimum value is 0.25, maximum value is 15 (realmin, realmax)
  - 48 positive numbers, 48 negative numbers
- If E=0, and M≠0, then value is $-1^S \times 2^{-2} \times 0.M$
  - 1/32, 2/32, 3/32, 4/32, 5/32, 6/32, 7/32 (and their negatives)
  - These are the unnormalized numbers (14 of them)
- If E=0 and M=0, then value is 0 or -0 (based on S)
- If E=7 and M=0, then value is Inf or –Inf (based on S)
- If E=7 and M≠0, then value is NaN (14 of these)

[0 000 011] =?
[1 000 110]= ?

Q3: If it is [0 001 110] , then Value =? (Decimal value =?)
Q4: if [1 110 110] / [1 010 011] / ..., Value ?