

國立臺北教育大學 101 學年度碩士班招生入學考試

數位科技設計學系玩具與遊戲設計碩士班 計算機概論 科試題

一、簡答題：(共 50 分)

- (一) 何謂電腦的五大單元(5 分)? 請繪圖說明電腦處理資料的流程圖(5 分)。
- (二) 請說明中央處理器(CPU)包含哪些裝置或元件(10 分)? 並請說明其處理與作動的方式(10 分)?
- (三) 何謂程序導向式設計(5 分)? 何謂物件導向程式設計(5 分)?
- (四) 何謂串流技術, 請列舉二個目前有支援串流技術的軟體, 這些軟體所支援的又是哪一種檔案格式。如果現在要你製作一個在網路上播放教學影片, 你會選擇哪一種格式, 請說明你選擇的理由。(10 分)

二、填充題：請將答案依照號碼寫在答案卷上, 務必註明題號與問題號碼。

(每格 2 分, 共 50 分)

- (一) 請完成以下對電腦硬體結構上的描述。我們說這顆 CPU 是 32 位元, 是由 CPU 中 (a) 的位元組數來決定。而 CPU 的時脈計算方式為 (b), 北橋負責連結 CPU 與 (c)、(d) 等高速的裝置, 而 (e) 則負責連結 CPU 與周邊低速的裝置。
- (二) 21.812510 轉換成 2 進位表示法為 (a), 轉換成 16 進位表示法為 (b) 另外以一個 Byte 為單位, -4510 轉換成 2 進位時, 以最高位元表示法表示時為 (c), 以 1 的補數表示法表示時為 (d), 以 2 的補數表示法表示時為 (e)。
- (三) 一個被稱為具有「結構化(structure)」的程式, 必須具備模組化的設計而且僅使用 (a)、(b) 與重複結構來完成。for、while 與 do...while 哪一個迴圈至少會執行迴圈主體一次 (c)。for(A; B; C)D;, 其中 A、

B、C、D 分別代表該部分的程式碼，假設這個 for 迴圈僅執行迴圈主體一次，它的執行順序為 (d)。

int i, j; j = 1; for(i = 0 ; i < 10 ; i++) j+=2; 執行完畢，i、j 的值各為 (e)。

(四) 下列程式碼計可輸入的數值是不是 2 的次方，請根據註解的說明，完成程式碼片段。

```
int iNum, iSave, iPower, iFlag;
```

```
(a); // iPower 儲存 2 的次方的次方值，設定 iPower 初始值
```

```
(b); // iFlag 代表目前的數是否為 2 的次方，設定 iFlag 初始值
```

```
printf("請輸入一個整數\n"); scanf("%d",&iNum);
```

```
iSave = iNum; /*保留使用者輸入，以便後面輸出*/
```

```
while ( (c) ) {
```

```
    if( iNum % 2 == 0 ) iPower++; //可以被 2 整除
```

```
    else (d); //不能被 2 整除
```

```
    (e); //對 iNum 執行必要的運算讓迴圈能正常執行
```

```
}
```

```
if( iFlag ) printf("你輸入的數值%d 是2 的%d 次方\n",iSave,iPower);
```

```
else printf("你輸入的數值%d 不是 2 的次方\n",iSave);
```

(五) 以下的程式碼是堆疊的實做程式，stack 為堆疊變數，請完成程式碼的內容，讓程式能順利的將字元 a 到 e 依序的儲存到堆疊中，然後再根據堆疊的規則依序的讀出 e 到 a。程式碼的內容必須處理堆疊空間不足，或是目前已經沒有資料可以取出的例外狀況。注意：假設堆疊是從位置 stack[0]開始儲存元素，而 sp 所指的位置永遠是堆疊目前可以被拿出元素的位置。

```
#define BUFSIZE 10 //定義陣列容量
```

```
char pop(void); //取出堆疊資料的函式
```

```
void push(char); //放入堆疊資料的函式
```

```
void dump(void); //顯示堆疊資料的函式
```

```
static char stack[BUFSIZE]; //以陣列模擬堆疊空間
```

```
static int sp; //堆疊頂端資料的位置變數
```

```

int main(void)
{
    int i; char c;
    sp = -1 ; // 設定 sp 的起始位置
    puts("Pushing...");
    for ( i = 0 ; i < 5 ; i++) { //控制將資料放入堆疊的迴圈
        push('a'+i); printf("放進%c ", 'a'+i); dump();
    }
    puts("\nPoping..."); //控制將資料取出堆疊的迴圈
    for ( i = 0 ; i < 5 ; i++) { c = pop(); printf("取出%c ", c); dump(); }
    return(0);
}

void push(char s) { // 放字元進堆疊
    if(__(a)__) //判斷堆疊是否已經滿了
        __(b)__; //放入元素同時修正 sp
    else printf("ERROR: stack full\n");
}

char pop(void) { //從堆疊取出字元
    char ra = 0;
    if(__(c)__) //判斷堆疊是否還有元素
        __(d)__; // 取出元素同時修正 sp
    else printf("ERROR: stack empty\n");
    return (ra);
}

void dump(void) { //由上到下，顯示堆疊中的所有資料
    int i; printf("堆疊內容:");
    if ( sp >= 0 ) //判斷堆疊是否有內容
        for(__(e)__) printf("%c ",stack[i]);
    else printf("EMPTY!");
}

```