

=====注意:以下 10 題作業只能引入 `stdio.h` 這個 header file=====

#### 1. 位元 1 的個數

目的：設計一組合語言副程式，宣告如下

```
unsigned int Count1Bits(unsigned int value);
```

計算 Value 內的位元為 '1' 的個數，回傳給主程式。

```
範例:int main(void)
{
    unsigned int Value=0x5AA555AA;
    unsigned int Count;

    Count=Count1Bits(Value);
    printf("Value=%08X, Count=%d\n", Value, Count);
    return 0;
}
```

#### 2. 找出 2 數的最大值(find max)

目的：設計一組合語言副程式，宣告如下

```
short _Max(short x, short y);
```

將 x 與 y 的最大值，回傳給主程式。

```
範例:int main(void)
{
    short x=0x1234, y=0x8001, max;
    max=_Max(x, y);
    printf("max of x, y = %d\n", max);
    return 0;
}
```

#### 3. 找出三數中的最小值(find min)

目的：設計一組合語言副程式，宣告如下

```
int _min(int x, int y, int z);
```

將 x, y 與 z 的最小值，回傳給主程式。

```
範例:int main(void)
{
    int n1, n2, n3, min;
    n1 = -2;      n2 = 0x1023;  n3 = 100;
    min = _min(n1, n2, n3);
    printf("min of x, y and z is %d\n", min);
    return 0;
}
```

#### 4. 資料總和(sum of data)

目的：設計一組合語言副程式，宣告如下

```
int Sum(int *x, int n);
```

計算一串數目之總和，此串數目的個數在 n 中，

數目本身由 x 開始。把此串數目之總和回傳給主程式。

```
範例:int main(void)
{
    int n=3; // 資料的個數
    int sum; // 儲存總和
    int x[]={2040,-1222,242}; // 資料由此開始
    sum = Sum(x, n);
    printf("Sum of x = %d\n", sum);
    return 0;
}
```

#### 5. 尋找最大值(find max number of an array)

目的：設計一組合語言副程式，宣告如下

```
int MAX(short *x, int n);
```

在一串數目中找出最大的元素。此串數目的元素個數在 n

中，資料本身由 x 開始，把最大的元素回傳給主程式。

```
範例:int main(void)
{
    int n = 4;      int _Max;
    short x[]={0xA484, 0x71AC, 0xE57A, 0x32F1};
    _Max = MAX(x, n);
    printf("Max of x is %04X\n", _Max);
    return 0;
}
```

#### 6. 正數的個數(count positive number)

目的：設計一組合語言副程式，宣告如下

```
int Positive(int *x, int n);
```

在一串數目中計算「正數」的個數。此串數目的個數存

於 n 中，且數目本身由 x 開始。把正數的個數回傳給主程式。

```
範例:int main(void)
{
    int n = 6,      int Pos;
    int Start[]={100, -101, 0x2120, 0, 0xE605, 4};
    Pos = Positive(Start, n);
    printf("Pos = %d\n", Pos);
    return 0;
}
```

=====

## 7. 零、正數與負數的個數

目的：設計一組合語言副程式，宣告如下

```
int ZeroCount(int *x, int n, int *pos, int *neg);
```

計算一串數目中，零、正數與負數的個數。此串數目的個數存

於 **n** 中，且數目本身由 **x** 開始。把正數與負數的

個數分別存於 **pos** 與 **neg** 為「指標」的記憶體中。

同時把『零』的個數回傳給主程式。

範例: 

```
int main(void)
```

```
{
    int n = 7;
    int Pos;      int Zero;      int Neg;
    int Start[]={7602, 848, -2120, 0, -605, 4, 0};

    Zero = ZeroCount(Start, n, &Pos, &Neg);
    printf("Zero=%d, Pos.=%d, Negative=%d\n", Zero, Pos, Neg);
    return 0;
}
```

=====

## 8. 字串長度

目的：設計一組合語言副程式，宣告如下

```
int len(char *buffer);
```

計算一串 **ASCII** 字串的長度。字串所存放的開始位址為 **buffer**，

字串的結束字元為 '\0'。將此字串的長度(字元數，不含結束

字元)求出，回傳給主程式。

範例: 

```
int main(void)
```

```
{
    char *Start="Hello, this is my first program!";
    int Len;
    Len = len(Start);
    printf("string=%s\n", Start);
    printf("length=%d\n", Len);
    return 0;
}
```

=====

## 9. 記憶體 copy

目的：設計一組合語言副程式，宣告如下

```
void copy(char *dest, char *src, unsigned int n);
```

從記憶體 **src** 開始 copy **n** 個 bytes 到記憶體 **dest** 去。

範例: 

```
int main(void)
```

```
{
    int len(char*); //宣告 len 函數
    char *src="Hello, this is my first program!";
    char dest[200];
    unsigned int n;

    n = len(src)+1; //呼叫第 8 題所設計的函數
    copy(dest, src, n);
    printf("src=%s\n", src);
    printf("dst=%s\n", dest);
    return 0;
}
```

=====

## 10. 1+2+...+n

目的：設計一組合語言副程式，宣告如下

```
int sum(int n);
```

計算 1+2+...+n (**n** >= 2) 把結果回傳給主程式。

必須使用迴圈編寫。(僅可以使用 **for**, **do**, **while** 迴圈)

範例: 

```
int main(void)
```

```
{
    int n=100, total;
    total = sum(n);
    printf("1+2+...+n=%d\n", n, total);
    return 0;
}
```

=====

「組合語言與系統程式」作業與驗收注意事項。

1. 驗收時不得攜帶任何參考資料或筆記。
2. 驗收時用抽籤決定題號。
3. 用 **arm** 組合語言編寫，於【學期結束】前一個星期驗收。

### 4. 成績計算方式

- (a) 期中考 **30%**(close books, notes)
- (b) 期末考 **40%**(close books, notes)
- (c) 出席 **10%**
- (d) 助教 **10%**
- (e) 期末作業驗收 **15%**(close books, notes)

黃連進 2017/09/28