

# 資料結構 期中考 (2018/11/13)

分數

系所：\_\_\_\_\_ 學號：\_\_\_\_\_ 姓名：\_\_\_\_\_

1. (10%) 寫一 C 副程式 `swap`，該副程式將傳入的兩個整數互換，例如執行下列範例後會列印 `x=2, y=1`。你寫的副程式 `swap` 必須能夠讓下列範例呼叫否則不計分。

```
void main( )
{
    int x=1, y=2;
    swap(&x,&y);
    printf("x=%d, y=%d \n", x , y );
    system("PAUSE");
}
```

## Answer 1

2. (10%) 執行下列程式螢幕會有什麼結果

```
void main()
{
    char X[10]={'a','b','3','c','4','\0'};
    char Y[10]={'5','6','x','y','7','8','z','\0'};
    char *p, *q,tmp;
    p=&X[0];
    q=Y;
    while(1) {
        while((*p!='\0')&&('a'<=*p)&&(*p<='z'))
            p++;
        if(*p=='\0')
            break;
        while((*q !='\0') &&('0'<=*q)&&(*q<='9'))
            q++;
        if(*q=='\0')
            break;
        tmp=*p;
        *p=*q;
        *q=tmp;
        p++;
        q++;
    }
    printf("%s\n",X);
    printf("%s\n",Y);
}
```

## Answer 2

3. (15%) 設計一副程式 **larger** 輸出兩個輸入較者。Larger 的輸入包含三個 **pointer**，其中兩個為所需比較大小之資料位址，另一個 **pointer** 實際比較大小的副程式。你所設計的副程式 **larger** 必須能提供下列程式呼叫，程式輸出為 2, 0.2。

```
int compare_int(void* ptr1, void* ptr2 )
{
    if( *(int *)ptr1 >=*(int *)ptr2 )
        return 1;
    return -1;
}
```

```
int compare_float(void* ptr1, void* ptr2 )
{
    if( *(float *)ptr1 >=*(float *)ptr2 )
        return 1;
    return -1;
}
```

```
void main( )
{
    int i1=1, i2=2, lrg_int;
    float f1=0.1, f2=0.2, lrg_float;
    lrg_int=*((int*)larger(&i1, &i2, compare_int));
    lrg_float=*((float*)larger(&f1, &f2,
compare_float));
    printf("%d, %f",lrg_int, lrg_float);
    system("PAUSE");
}
```

### Answer 3

4. 請回答以下問題

```
void bs(int list[], int n)
{
    int temp, current, sorted, walker;
    sorted=0;
    for(current=0; current<=n && !sorted; current++)
    {
        sorted=1;
        for(walker=n; walker>current; walker--)
            if(list[walker] < list[walker-1])
            {
                sorted=0;
                temp=list[walker];
                list[walker]=list[walker-1];
                list[walker-1]=temp;
            }
    }
}
```

```
void main( )
{
    int list[10]={1,3,5,7,9,2,4,6,8,10};
    int i;
    bs(list, 10);
    for(i=0;i<10;i++)
        printf("(%d)",list[i]);
    system("PAUSE");
}
```

- (10%)執行此程式後輸出為何?
- (10%)請分析並寫出副程式 **bs** 的 Big-O.

### Answer 4

a.	b.

5. (15%) 設計一程式使用 **recursive** 技巧讓使用者輸入一串整數，程式輸入負整數時將該負整數之前的非負整數以相反的順序列印。

例如：

若輸入

1

3

5

-9

則輸出

5

3

1

### Answer 5

6. (15%)

```
typedef struct node
{
    void* dataPtr;
    struct node* next;
} QUEUE_NODE;
```

```
typedef struct
{
    QUEUE_NODE* front;
    QUEUE_NODE* rear;
    int count;
} QUEUE;
```

```
int enqueue (QUEUE* queue, void* itemPtr )
{
    QUEUE_NODE* newPtr;
    if(!(newPtr=(QUEUE_NODE*)malloc(sizeof(QUEUE_NODE))))
        return 0; //false;
    newPtr->dataPtr=itemPtr;
    newPtr->next=NULL;
```

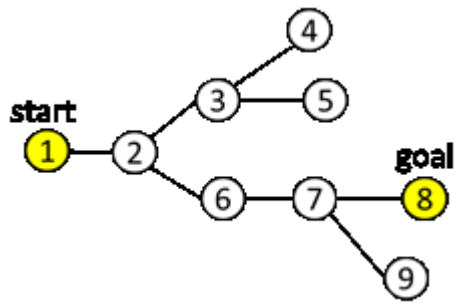
//課本中為 6 行程式碼，但你可以填入超過 6 行程式碼，只要功能正確即可

```
    return 1;
}
```

方框內填入數行程式碼以完成 **enqueue** 動作，其中 **itemPtr** 為指向欲加入該 **queue** 的資料的位址。

### Answer 6

7. (15%) 說明如何使用 **stack** 找出由起始點 1 到終點 8 的路徑，需解釋每個步驟所用到的 **stack** 運作(例如 **push**, **pop**)並繪出 **stack** 內容。



**Answer 7**