



## ข้อที่ 1 ถ้าถามวัดความรู้พื้นฐาน

(30 คะแนน)

1.1 จากอาร์เรย์  $a$  จงเขียนส่วนของโค้ด เพื่อหาผลรวมของสมาชิกที่เป็นเลขคู่ทุกตัว โดยนำผลลัพธ์เก็บไว้ในตัวแปร  $total$

```
int a[5] = {3, 2, 7, -4, 6};
```

1.2 จงเขียนส่วนของโค้ด เพื่อสร้างชนิดข้อมูลใหม่ ประเภท enum โดยให้ชื่อว่า `traffic_light` ซึ่งมีค่าความเป็นไปได้ของสี คือ RED, YELLOW, GREEN พร้อมทั้งประกาศตัวแปร  $t1$  ซึ่งมีค่าเริ่มต้นเป็น RED

1.3 จากการเรียกใช้งานฟังก์ชันข้างล่าง จงเขียนต้นแบบของฟังก์ชันที่เหมาะสม

```
foo(foo('a', 2), 1);
```

1.4 จงเขียนต้นแบบของฟังก์ชันที่เหมาะสมกับการใช้งานดังต่อไปนี้

- เปลี่ยนอักษรตัวเล็กเป็นอักษรตัวใหญ่
- เปลี่ยนสัญญาณไฟจราจรเป็นสัญญาณไฟตัดไป (ให้ใช้การนิยามโครงสร้างในข้อ 1.2)
- หาผลบวกของสมาชิกทั้งหมดในอาร์เรย์ของ `double`
- กำหนดให้สมาชิกทุกตัวในอาร์เรย์ของ `int` เป็น 0
- หาตำแหน่งในหน่วยความจำของ สมาชิกตัวแรก ในอาร์เรย์ของ `int` ที่มีค่าเป็น 0

1.5 จากตัวแปรพอยเตอร์  $ptr$  และตัวแปรเลขจำนวนเต็ม  $x$  จงใช้พอยเตอร์  $ptr$  เปลี่ยนค่าของ  $x$  ให้เป็น 10

```
int x = 0;  
int *ptr;
```

1.6 จงเขียนส่วนของโค้ดที่ใช้พอยเตอร์ iptr และ Dereferencing แทน [] ที่ให้ผลแบบเดียวกันกับโค้ดข้างล่าง

```
int i;
int *iptr;
char str[] = "Hello";

//rewrite your code here
for(i = 0; str[i] != '\0'; i++)
    printf("%d", str[i]);
```

1.7 จากโค้ดที่กำหนดให้ จงเขียนส่วนของโค้ด เพื่อใช้พอยเตอร์ ptr เปลี่ยนแปลงให้ค่าเริ่มต้นกับ e โดยให้ name มีค่าเป็น "John Doe" และมีค่าเงินเดือนเป็น 25000 บาท

```
typedef struct {
    char name[128];
    double salary;
} Employee;
Employee e;
Employee *ptr;
```

## ข้อที่ 2 ข้อความและตัวอักษร

(30 คะแนน)

จงเขียนฟังก์ชันเพื่อทำการเข้ารหัสข้อความ โดยหลักการของการเข้ารหัส คือตัวหนังสือจะเปลี่ยนเป็นตัวหนังสือตัวถัดไป ตัวอย่างเช่น dog จะเข้ารหัสเป็น eph หรือ cat จะเข้ารหัสเป็น dbu เป็นต้น โดยฟังก์ชันดังกล่าวจะรับพารามิเตอร์เป็น char[] และให้ผลลัพธ์เป็นข้อความใหม่ที่ทำการเข้ารหัสใหม่แล้ว

ทั้งนี้ฟังก์ชันจะต้องสนับสนุนการทำงานดังแสดงข้างล่าง

- หากตัวอักษรเริ่มต้นเป็นตัว z ให้ตัวที่เข้ารหัสแล้วเป็นตัว a
- หากตัวอักษรเริ่มต้นเป็นตัวพิมพ์ใหญ่ ให้ตัวอักษรที่เข้ารหัสเป็นตัวพิมพ์ใหญ่ตัวถัดไป (ใช้หลักการเดียวกับตัวพิมพ์เล็ก)
- หากตัวอักษรเริ่มต้นเป็นตัวอักษรที่ไม่ใช่ตัวอักษร เช่นตัวเลข หรืออักขระพิเศษ ให้คงไว้เป็นตัวอักษระเดิม
- ต้นแบบของฟังก์ชันคือ

```
char[] encrypt(char[]);
```

### ข้อที่ 3 อาร์เรย์ พอยเตอร์และหน่วยความจำ

(40 คะแนน)

โค้ดข้างล่าง เป็นกำหนดให้เป็นการหาค่า Transpose ของเมตริกซ์ที่กำหนดไว้ในอาร์เรย์สองมิติ a โดยผลลัพธ์ของการรันโปรแกรม (เมื่อเติมให้สมบูรณ์แล้ว) คือ

```
1 2 3
7 4 8
```

```
1 #include <stdio.h>
2
3 int* transpose(int*, int, int);
4
5 void main() {
6     int a[3][2] = {
7         {1, 7},
8         {2, 4},
9         {3, 8}
10    };
11    int *b, *t;
12    int i, j;
13
14    b = transpose(a, 3, 2);
15
16    t = b;
17
18    /** add your code here */
19 }
20
21 int* transpose(int *m, int row, int col) {
22     int *r = (int*) malloc(sizeof (int) * row * col);
23     int *t = r;
24     int i, j;
25     for (j = 0; j < col; j++)
26         for (i = 0; i < row; i++)
27             *(t++) = *(m + j + col * i);
28     return r;
29 }
```

3.1 จงอธิบายการทำงานของฟังก์ชัน transpose ในบรรทัดที่ 22, 23, 25-27, 28 ให้ใช้แผนภาพประกอบคำอธิบาย ในจุดที่เป็นไปได้ พร้อมทั้งให้เหตุผลว่า ทำไมการใช้ฟังก์ชัน malloc จึงเป็นสิ่งจำเป็นในฟังก์ชัน transpose

3.2 จงเติมส่วนของโค้ดที่ขาดหายไปในช่วงบรรทัดที่ 18 เพื่อให้โปรแกรมสมบูรณ์

#### ข้อที่ 4 โครงสร้างข้อมูล

(50 คะแนน)

จากโปรแกรมที่กำหนดให้ จงเขียนนิยามของฟังก์ชันทั้งสาม เพื่อให้โปรแกรม แสดงผลลัพธ์ดังนี้

```
John has 14400.0 baht a month.  
Mark has 30000.0 baht a month.  
Clark has 24000.0 baht a month.
```

```
1 #include <stdio.h>  
2 #include <stdlib.h>  
3  
4 typedef struct  
5     char name[128];  
6     float salary;  
7 } Employee;  
8  
9 float salary_per_level[] = {12000, 15000, 20000, 30000, 35000};  
10  
11 void e_init(Employee *e, char* name, int level);  
12 void e_increase_salary(Employee list[], int indexes[],  
13     int len, float percent);  
14 void e_display(Employee *e);  
15  
16 void main() {  
17     Employee list[3];  
18     int indexes[] = {0,2};  
19     int i;  
20  
21     e_init(&list[0], "John", 0);  
22     e_init(&list[1], "Mark", 3);  
23     e_init(&list[2], "Clark", 2);  
24  
25     e_increase_salary(list, indexes, 2, 0.2);  
26  
27     for(i = 0; i < 3; i++)  
28         e_display(list + i);  
29 }  
30
```

4.1 จงเขียนนิยามของฟังก์ชัน `e_init` ให้ทำการกำหนดค่าเริ่มต้นให้กับ `e` โดยเงินเดือนเริ่มต้นจะใช้ `level` เป็นอินเด็กซ์ของ `salary_per_level` เพื่อหาเงินเดือนเริ่มต้นสำหรับแต่ละระดับ เช่น นาย John มี `level` เป็น 0 จะมี `salary` เริ่มต้นเป็น 12000 บาท

4.2 จงเขียนนิยามของฟังก์ชัน `e_increase_salary` โดยจะทำการเพิ่มเฉพาะสมาชิกที่มีหมายเลขอินเด็กซ์ระบุไว้ใน `indexes` ทั้งนี้ `len` จะกำหนดว่ามี `indexes` จำนวนกี่ตัว การเพิ่มเงินเดือนจะเพิ่มจากเดิมเป็นเปอร์เซ็นต์เท่ากับค่าตัวแปร `percent`

4.3 จงเขียนนิยามของฟังก์ชัน `e_display` เพื่อแสดงผล `e` (การแสดงผลจะต้องสอดคล้องกับผลลัพธ์ของโปรแกรม)

### ข้อที่ 5 ประยุกต์ใช้

(30 คะแนน)

เพื่อหาแนวทางรับมือ สงครามอาวุธชีวภาพ หน่วยงานลับได้คิดค้นเซ็นเซอร์ที่สามารถตรวจจับเชื้อโรคต่างๆ จากอาวุธชีวภาพ โดยจะวัดระดับของเชื้อโรคจาก 0-100 โดยในแผนการตั้งรับระบุไว้ว่า ระหว่างสงคราม จะโปรยเซ็นเซอร์ชนิดนี้ทางอากาศ โดยจะทำการโปรยเป็นโซนทั้งหมด 50 โซน โซนละ 1000 ตัว

ท่านได้รับมอบหมายให้เขียนโปรแกรม เพื่อบรรจุลงไปในอุปกรณ์ตรวจจับเชื้อโรค ภายใต้โค้ดเนม `Werewolf` ที่จะถูกพกพาโดยทหาร เพื่อได้รับการแจ้งเตือนเมื่อเข้าสู่เขตเชื้อโรค เพื่อตรวจสอบระดับเชื้อโรค เจ้าหน้าที่จะกดปุ่มเพื่ออ่านข้อมูลจากเซ็นเซอร์ทั้ง 1000 ตัว ในโซนนั้น เครื่องจะทำการหาค่าเฉลี่ยของระดับเชื้อโรคจากเซ็นเซอร์ทุกตัวในโซนนั้น แล้วแสดงผลบนหน้าจอของอุปกรณ์ตรวจจับ

ท่านได้รับข้อมูลเพิ่มเติมดังต่อไปนี้

- เพื่อให้มั่นใจว่าค่าข้อมูลเซ็นเซอร์ถูกต้อง หลังจากการอ่านค่าแล้ว จะต้องตรวจสอบด้วย `validate_sensor` หากข้อมูลไม่ถูกต้อง ให้ตัดข้อมูลตัวนี้จากการคำนวณ
- หากมีเซ็นเซอร์เสียไปมากกว่า 50 เปอร์เซ็นต์ ให้อุปกรณ์แสดงค่าว่าไม่สามารถอ่านค่าได้
- ในการควบคุมอุปกรณ์ฮาร์ดแวร์บนอุปกรณ์ตรวจจับ ให้ใช้ผ่านฟังก์ชันต่างๆ ใน `werewolf.h` (ให้ถือว่าทุกฟังก์ชันถูกนิยามไว้เรียบร้อยแล้ว)
- อุปกรณ์จะเรียกฟังก์ชัน `main` อัตโนมัติ เมื่อทหารกดปุ่มเพื่อทำการอ่านข้อมูล

```

/** werewolf.h */

/* ค้นหาว่า บัจุบันอุปกรณ์อยู่ใน zone หมายเลขใด เนื่องจากฟังก์ชันนี้มีการใช้สัญญาณดาวเทียม ควรถูกเรียกใช้
น้อยครั้งที่สุด เพื่อป้องกันการตรวจพบโดยศัตรู */
int lookup_zone();

/* อ่านค่าจากเซ็นเซอร์ตัวที่กำหนด */
int read_senser(int zone, int sensor_no);

/* รอจนกว่าเครื่องอ่านจะพร้อมสำหรับการ validate ข้อมูล เนื่องจากฟังก์ชันนี้จะต้องใช้พลังงานของเครื่องอ่าน
มาก จึงควรถูกเรียกใช้น้อยครั้งที่สุด */
void prepare_validation();

/* ตรวจสอบข้อมูลที่อ่านได้ หากข้อมูลผิดปกติ ข้อมูลจะถูกเปลี่ยนค่าเป็น -1 ก่อนเรียกใช้ฟังก์ชันนี้เครื่องจะต้องพร้อม
สำหรับการ valitdate โดยการใ้ฟังก์ชัน prepare_validation */
void validate_sensor(int zone, int sensor_no, int *value) ;

/* แสดงค่าระดับเชื้อโรคที่อ่านได้บนหน้าจอ หาก level เป็น -1 เครื่องอ่านจะแสดงผลว่า ไม่สามารถอ่านค่าได้ */
void display_bio_threat(int level);

```