มหาวิทยาลัยสงขลานครินทร์
คณะวิศวกรรมศาสตร์

สอบกลางภาค: ภาคการศึกษาที่ 1 **ปีการศึกษา**: 2550

วันที่สอบ: 30 กรกฎาคม 2550 **เวลาสอบ**: 09.00 – 12.00 น.

รหัสวิชา: 241-207, 240-204

ห้องสอบ: EE213, R200, R201

ชื่อวิชา: Data Structure and Computer Programming Techniques

ทุจริตในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ เข้าห้องสอบ

ไม่อนุญาต: หนังสือ หรือเครื่องคิดเลขเข้าห้องสอบ และ**เอกสารใด ๆ เข้าและออกห้องสอบ**

เวลา: 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ข้อสอบมี 11 หน้า (รวมใบปะหน้า) รวมทั้งหมด 7 ข้อ คิดเป็นคะแนนเก็บ 35 %
- **<u>คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด</u>**
- อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- หากข้อใดเขียนตอบไม่พอ ให้เขียนเพิ่มที่ด้านหลังของหน้านั้นเท่านั้น

ชื่อ_____**รหัสนักศึกษา**_____**ตอน**____

รหัสนักศึกษา_____ตอน_____

ตัวอย่าง Function prototypes ประกอบการทำข้อสอบ

<string.h>

```
char *strcpy(char *s1, const char *s2);
// copy s2 -> s1      returns s1

char *strcat(char *s1, const char *s2);
// concatenates s2->s1      returns s1

int strcmp(const char *s1, const char *s2);
//returns    < 0    if s1 < s2
               0    if s1 == s2
             > 0    if s1 > s2

char *strchr(const char *str, int ch);
/* The function strchr() returns a pointer to the first occurence of ch
in str, or NULL if ch is not found. */

size_t strlen(const char *str);
/* The strlen() function returns the length of str (determined by the
number of characters before null termination).  */
```

<stdlib.h>

```
void *calloc( size_t num, size_t size );

/* The calloc() function returns a pointer to space for an array of
num objects, each of size size. calloc() returns NULL if there is an
error. */

void *malloc( size_t size );
/* The function malloc() returns a pointer to a chunk of memory of
size size, or NULL if there is an error.  */
```

รหัสนักศึกษา_____ตอน_____

1. จงอธิบายโปรแกรมด้านล่างว่า**ผลการทำงานเป็นอย่างไร** โดย**อธิบายหรือวาดรูปประกอบ**ว่าในการเรียกใช้งานฟังก์ชัน aFunction() ในแต่ละรอบ **ค่าพารามิเตอร์ที่ให้กับฟังก์ชันคือค่าใดบ้าง** (2 คะแนน)

```
#include <stdio.h>
#define SIZE 5
void aFunction(int [], int);

int main()
{
  int myArray[SIZE] = {32, 27, 64, 18, 95};

  printf("The values in the array are: ");
  aFunction(myArray, SIZE);
  return 0;
}

void aFunction(int a[], int size)
{
  if (size > 0) {
    aFunction(&a[1], size-1);
    printf("%d ", a[0]);
  }
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

รหัสนักศึกษา_____ตอน_____

2. **จงเขียนโปรแกรม**รับตัวเลขจำนวนเต็มเข้ามา 1 ค่า แล้ว**ตรวจสอบว่าตัวเลขที่รับเข้ามาอยู่ระหว่าง 65-122 หรือไม่** ถ้าไม่ใช่ให้วนรับค่าใหม่ จากนั้นให้โปรแกรม**พิมพ์ตัวอักษรที่มีรหัส** ASCII **ตรงกับตัวเลขที่รับเข้ามา** (2 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. จงเขียน**ผลลัพธ์ของการทำงานของโปรแกรม พร้อมทั้งวาดภาพการเก็บค่าในหน่วยความจำหรือวาดลูกศรว่า** Pointer **ชี้ไปที่ตำแหน่งตัวแปรใด** (2 คะแนน)

```
#include<stdio.h>
int main() {

      int x = 1, y = 1, *p, *q;

      p = &x;  q = &y;  x = 7;        // ----------------> (2.1)
      printf("%d %d %d %d \n", x, y, *p, *q);

      *q = 5;  x = 9;  p = q;         // ----------------> (2.2)
      printf("%d %d %d %d \n", x, y, *p, *q);

      *p = *q+3;                      // ----------------> (2.3)
      printf("%d %d %d %d \n", x, y, *p, *q);

      if(p == q)     printf("The pointer are equal \n");
      if(*p == *q)   printf("The objects are equal \n");
      return 0;
}
```

3.1 ผลการรันโปรแกรม // 0.5 คะแนน

.....

.....

.....

.....

.....

รหัสนักศึกษา_____ตอน_____

3.2 การเก็บค่าในหน่วยความจำของตัวแปร x, y, p และ q หลังจบบรรทัดที่ 2.1 // 0.5 คะแนน

☐ X ☐ y

☐ p ☐ q

3.3 การเก็บค่าในหน่วยความจำของตัวแปร x, y, p และ q หลังจบบรรทัดที่ 2.2 // 0.5 คะแนน

☐ X ☐ y

☐ p ☐ q

3.4 การเก็บค่าในหน่วยความจำของตัวแปร x, y, p และ q หลังจบบรรทัดที่ 2.3 // 0.5 คะแนน

☐ X ☐ y

☐ p ☐ q

4. ใช้โปรแกรมด้านล่างตอบคำถามต่อไปนี้ (5 คะแนน)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
struct card {
   char  *face;
   char  *suit;
};
typedef struct card Card;
void filldeck(Card[], char*[], char*[]);
int main()
{
   Card deck[52];
   char *face[] = {"Ace", "Deuce", "Three", "Four", "Five", "Six", "Seven",
                   "Eight", "Nine", "Ten", "Jack", "Queen", "King"};
   char *suit[] = {"Hearts", "Diamonds", "Clubs", "Spades"};
   filldeck(deck, face, suit);
   return 0;
}
```

รหัสนักศึกษา_____ตอน_____

4.1 คำสั่ง typedef **มีไว้เพื่ออะไร** // 1 คะแนน

.....

.....

.....

4.2 **จงเติมฟังก์ชัน** filldeck(Card [ ], char *[ ], char *[ ]) **ให้สมบูรณ์** โดยฟังก์ชัน filldeck() มีไว้เพื่อกำหนดค่าเริ่มต้นให้กับตัวแปร Card deck[52] โดยที่แต่ละอีลีเมนต์ในตัวแปร deck[ ] จะต้องมีค่าไม่ซ้ำกัน (ไพ่แต่ละตัวมีลักษณะเฉพาะ ไม่ซ้ำกับไพ่ใบอื่นในสำรับเดียวกัน) // 4 คะแนน

```
void filldeck(Card wdeck[], char *wface[], char *wsuit[])
{
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
}
```

5. **เขียนโปรแกรม**ที่รับข้อมูลจากผู้ใช้ 2 ค่าคือ (1) รับข้อความเข้ามาหนึ่งบรรทัด และ (2) รับคำเข้ามาหนึ่งคำ จากนั้นตรวจสอบว่า **คำที่รับเข้ามาปรากฏอยู่ในข้อความเป็นจำนวนกี่ครั้ง** (10 คะแนน)

ตัวอย่างผลการรันโปรแกรม (อักษรตัวเอนเป็นค่าที่รับจากผู้ใช้)

Enter a sentence: *the cat sat on the mat*

Enter a word: *the*

The word occurs 2 times

จงเขียนอธิบายด้วยว่า **โปรแกรมที่นักศึกษาเขียนขึ้นนี้ มีขั้นตอนการทำงานหรือแนวคิดอย่างไร**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

รหัสนักศึกษา_____ตอน_____

6. **จงเติมโปรแกรมต่อไปนี้ให้สมบูรณ์** โปรแกรม bSort ทำงานโดยเรียงลำดับตัวเลขใน array ด้วยวิธี bubble sort จากน้อยไปหามาก (ascending) ยกเว้นเมื่อเรียกใช้งานโปรแกรมนี้ด้วย option –r (พิมพ์ –r ต่อจากชื่อโปรแกรมเมื่อเรียกใช้งานใน command prompt ของ unix) โปรแกรมจะเรียงลำดับตัวเลขใน array จากมากไปหาน้อย (descending) (10 คะแนน)

ตัวอย่างผลการรันโปรแกรม

```
$ bSort
1 2 3 5 8 9 35 45 77 90
$ bSort -r
90 77 45 35 9 8 5 3 2 1
```

```
#include<stdio.h>
#define SIZE 10

void bubble(int [], int (*)(int, int));
void swap(int*, int*);
int ascending(int, int);
int descending(int, int);

int main(int argc, int *argv[])  // 4 คะแนน
{
   int a[SIZE] = {2,5,1,9,35,77,8,90,3,45};

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
   return 0;
}
```

รหัสนักศึกษา_____ตอน_____

```
void bubble(int work[], int (*compare)(int, int))    // 3 คะแนน
{
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
}

int ascending(int a, int b)      // 0.5 คะแนน
{
    ..................................................................
    ..................................................................
}

int descending(int a, int b)     // 0.5 คะแนน
{
    ..................................................................
    ..................................................................
}

void swap(int *x, int *y)     /* swap x and y */      // 2 คะแนน
{
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
    ..................................................................
}
```

รหัสนักศึกษา_____ตอน_____

7. โปรแกรมต่อไปนี้ อาจให้ผลการทำงานที่ผิดพลาดเพราะอะไร จงบอกวิธีแก้ไขมา 2 วิธี (4 คะแนน)

```
struct card{
      char face;
      char suit;
};

int main()
{     struct card a;

      a=initialise();
      printf("%s", a.face);
            : // ชุดคำสั่งต่างๆ
      printf("%s", a.face);
      return 0;
}

struct card initialise(void)
{     struct card b;
      b.face="Three";
      return b;
}
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....