

PRINCE OF SONGKLA UNIVERSITY
FACULTY OF ENGINEERING



Midterm Examination: ภาคการศึกษาที่ 2

Academic Year: 2548

Subject Number: 240-204

Subject Title: Data Structures and Computer Programming Techniques

ทฤษฎีในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทฤษฎี และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รายละเอียดของข้อสอบ:

เวลา 3 ชั่วโมง (180 คะแนน: 180 นาที)

ข้อสอบมีทั้งหมด 9 หน้า (ไม่รวมหน้านี้) ประกอบด้วยคำถามจำนวน 3 ข้อ

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ

ไม่อนุญาต: หนังสือ และเครื่องคิดเลข

คำแนะนำ:

- พยายามทำทุกข้อ
- คำตอบทั้งหมดจะต้องเขียนลงในช่องว่างที่เว้นไว้ในข้อสอบ
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า เขียนโปรแกรมบางส่วน เขียนฟังก์ชัน หรือเขียนทั้งโปรแกรม รวมไปถึงข้อกำหนดเพิ่มเติม และหมายเหตุในข้อนั้น ๆ
- การเขียนโปรแกรมในแต่ละข้อ อาจจะไม่ต้องเขียนตามคำสั่งย่อยทั้งหมด แต่คะแนนจะลดลงตามส่วน และหากในข้อใหญ่หนึ่งข้อ นักศึกษาไม่สามารถทำข้อย่อยข้อแรก ๆ ได้ นักศึกษาสามารถทำข้อย่อยหลัง ๆ โดยให้อ้างอิงเหมือนนักศึกษาทำข้อย่อยข้อแรก ๆ ได้
- การเขียน code จะต้องตั้งชื่อตัวแปรให้เหมาะสม และมี comment ในจุดสำคัญต่าง ๆ โดยให้ทั้งหมดเป็นไปตามหลักการเขียนโปรแกรมที่ดี

ข้อที่ 1 WARM UP**(50 คะแนน: 50 นาที)**

- 1.1 ส่วนของโปรแกรมที่อยู่ด้านล่างนี้เป็นโค้ดของฟังก์ชัน `sum_1_to_n` โดยฟังก์ชันนี้จะ return ค่าผลบวกของจำนวนเต็มตั้งแต่ 1 ถึง `n` จงเติมโค้ดในส่วนบรรทัดที่ 3 และ 5 เพื่อให้ฟังก์ชันนี้ทำงานได้ตามที่กำหนด และจะต้องเขียนโค้ดให้ฟังก์ชันนี้เป็น Recursive Function

(15 คะแนน)

```

1.   int sum_1_to_n(int n) {
2.       if(n <= 1)
3.           .....
4.       else
5.           .....
6.
7.   }
```

- 1.2 โปรแกรมที่อยู่ด้านล่างนี้มีบางส่วนที่ขาดหายไป ซึ่งก็คือโค้ดในฟังก์ชัน `square_and_cube` โดยฟังก์ชันนี้ทำหน้าที่ในการหาค่าของการยกกำลังสองและการยกกำลังสามของพหามิเตอร์ `n` จงเติมโค้ดในบรรทัดที่ 16 และ 17 เพื่อให้ได้ผลการรันโปรแกรมตามที่กำหนด (10 คะแนน)

```

1.   #include <iostream>
2.   using std::cout;
3.
4.   void square_and_cube(int *sq, int *cb, int n);
5.
6.   int main() {
7.       int number = 10;
8.       int sq_val, cb_val;
9.       square_and_cube(&sq_val, &cb_val, number);
10.      cout << "Square of " << number << " is " << sq_val << "\n";
11.      cout << "Cube of " << number << " is " << cb_val << "\n";
12.      return 0;
13.  }
14.
15.  void square_and_cube(int *sq, int *cb, int n) {
16.      .....
17.      .....
18.  }
```

ผลการรันโปรแกรม

Square of 10 is 100
Cube of 10 is 1000

1.3 จากโปรแกรมที่อยู่ด้านล่างนี้ ฟังก์ชัน `sum_arr` ทำหน้าที่ในการคำนวณผลบวกของทุกอีลีเมนต์ในอาร์เรย์ จงเติมโค้ดในบรรทัดที่ 9 เพื่อเรียกใช้ฟังก์ชัน `sum_arr` นี้ และทำให้โปรแกรมสามารถแสดงผลรวมของทุกอีลีเมนต์ในอาร์เรย์ `numbers` ได้อย่างถูกต้อง (10 คะแนน)

```
1. #include <iostream>
2. using std::cout;
3.
4. int sum_arr(int arr[], int size);
5.
6. int main() {
7.     int numbers[] = {5, 8, 2, 1, 9};
8.     int total;
9.
10.     .....
11.     cout << "Sum of all elements is " << total << "\n";
12.     return 0;
13. }
14. int sum_arr(int arr[], int size) {
15.     int sum = 0;
16.     for(int i=0; i<size; i++)
17.         sum = sum + arr[i];
18.
19.     return sum;
20. }
```

ผลการรันโปรแกรม

Sum of all elements is 25

1.4 จงเติมโค้ดในบรรทัดที่ 27 เพื่อให้ได้ผลการรันโปรแกรมตามที่กำหนด

(15 คะแนน)

```
1. #include <iostream>
2. #include <cstring>
3. using std::cout;
4.
5. #define MAX_LEN 50
6.
7. class Student {
8. private:
9.     char std_name[MAX_LEN];
10.    int std_id;
11. public:
12.     Student(char *name, int id);
13.     void show_detail();
14. };
15.
16. Student::Student(char *name, int id) {
17.     strcpy(std_name, name);
18.     std_id = id;
19. }
20.
21. void Student::show_detail() {
22.     cout << "Student ID: " << std_id << "\n";
23.     cout << "Name: " << std_name << "\n";
24. }
25.
26. int main() {
27.     .....
28.     stud.show_detail();
29.     return 0;
30. }
```

ผลการรันโปรแกรม

Student ID: 4110005

Name: Somchai

ข้อที่ 2 ARRAYS & POINTERS

(40 คะแนน: 40 นาที)

2.1 จงเขียนฟังก์ชันชื่อ `sum_even` ซึ่งมี Prototype ดังนี้

```
int sum_even(int numbers[], int size);
```

โดยฟังก์ชัน `sum_even` นี้จะต้องสามารถหาค่าผลบวกของจำนวนคู่ทุกตัวที่อยู่ในอาร์เรย์ `numbers` และ `return` ค่าผลบวกดังกล่าว โดยที่พารามิเตอร์ `size` นั้นเป็นขนาดของอาร์เรย์

(20 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2.2 จงเขียนฟังก์ชันชื่อ `is_all_lower` ซึ่งมี Prototype ดังนี้

```
int is_all_lower (char *str);
```

ฟังก์ชัน `is_all_lower` นี้จะต้องสามารถตรวจสอบได้ว่าสตริงที่ `str` ซึ่อยู่นั้นเป็นตัวอักษรตัวเล็กทั้งหมดหรือไม่ ถ้าใช่ให้ return ค่าเป็น 1 แต่ถ้าไม่ใช่ให้ return ค่าเป็น 0

คำแนะนำ: ในการตรวจสอบว่าตัวอักษรเป็นตัวเล็กหรือไม่ สามารถเรียกใช้ฟังก์ชัน `islower` ใน Standard Library ได้เลย ซึ่งฟังก์ชัน `islower` มี Prototype ดังนี้

```
int islower (char c);
```

โดยจะ return ค่าเป็น 1 ถ้า `c` เป็นตัวอักษรตัวเล็ก และ return ค่าเป็น 0 ถ้า `c` ไม่ใช่ตัวอักษรตัวเล็ก
(20 คะแนน)

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

ข้อที่ 3 OOP**(90 คะแนน: 90 นาที)**

จงใช้ข้อมูลที่กำหนดให้ตอบคำถามข้อ 3.1-3.5

โค้ดข้างล่าง แสดง MyString.h และ Main.cpp โดยมีรายละเอียดดังนี้

MyString.h อธิบายต้นแบบของ class MyString ซึ่งทำหน้าที่ในการเก็บข้อความจำนวน 1 ข้อความไว้ และมีฟังก์ชัน getLength() ในการค้นหาความยาว และฟังก์ชัน append() ที่ใช้ในการต่อข้อความ โดย MyString สนับสนุนฟังก์ชัน undo() ที่ใช้ในการยกเลิกการแก้ไขข้อความครั้งล่าสุด เพื่อสนับสนุนการยกเลิกดังกล่าว class MyString จะมีข้อความ 2 อัน โดยเมื่ออันหนึ่งเก็บข้อความล่าสุด อีกอันหนึ่งจะเก็บข้อความก่อนการแก้ไขเอาไว้ ตัวแปร index ถูกใช้สำหรับการกำหนดข้อความล่าสุด

ทั้งนี้ MyString.h ข้างล่างยังแสดงนิยามของฟังก์ชัน getString() ไว้ด้วย

```
//-----MyString.h-----
#ifndef MYSTRING_H
#define MYSTRING_H

#include <cstring>

#define MAX_LEN 1024

class MyString{
private:
    char buf[2][MAX_LEN + 1];
    int index;
    const char* append(const char* str);
public:
    MyString(const char* str);
    const char* getString();
    int getLength();
    const char* append(MyString my_str);
    void undo();
};

const char* MyString::getString(){
    return (char*) buf[index];
}

#endif
```

หมายเหตุ นักศึกษาสามารถเรียกใช้ฟังก์ชันในการประมวลผลข้อความจาก string.h ได้

```
char *strcat(char *s1, const char *s2);
int strcmp(const char *s1, const char *s2);
char *strcpy(char *s1, const char *s2);
size_t strlen(const char *s);
char *strncat(char *s1, const char *s2, size_t n);
int strncmp(const char *s1, const char *s2, size_t n);
char *strncpy(char *s1, const char *s2, size_t n);
char *strtok(char *s1, const char *s2);
```

Main.cpp แสดงการใช้งาน MyString โดยมีผลการรันดังแสดงข้างล่างโค้ด

```
//-----Main.cpp-----
#include <iostream>
#include "MyString.h"

using std::cout;
using std::endl;

int main()
{
    MyString str1("Hello");
    MyString str2("World");
    cout << str2.getString() << " has " << str2.getLength()
        << " characters." << endl;

    str1.append(str2);
    cout << "After appending string 1 is "
        << str1.getString() << endl;

    str1.undo();
    cout << "Afer undoing string 1 is "
        << str1.getString() << endl;

    return 0;
}

=====Running Results=====
World has 5 characters.
After appending string 1 is HelloWorld
Afer undoing string 1 is Hello
```

3.1 จงเขียนนิยาม Constructor ของ class MyString อย่างเหมาะสม (20 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3.2 จงเขียนนิยามของฟังก์ชัน `append()` ทั้งสองฟังก์ชัน ของ class `MyString` โดยให้มีการเรียกใช้กันอย่างเหมาะสม (40 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3.3 เขียนนิยามของฟังก์ชัน `undo()` ของ class `MyString` อย่างเหมาะสม (10 คะแนน)

.....

.....

.....

.....

.....

.....

3.4 เขียนนิยามของฟังก์ชัน `getLength()` ของ class `MyString` อย่างเหมาะสม (10 คะแนน)

.....

.....

.....

.....

.....

.....

3.4 จงให้เหตุผลสนับสนุนการกำหนดฟังก์ชัน `append()` ของ class `MyString` เฉพาะที่มี
 ต้นแบบดังแสดงข้างล่างให้เป็น `private` (10 คะแนน)

```
const char* append(const char* str);
```

.....

.....

.....

.....

.....

.....