มหาวิทยาลัยสงขลานครินทร์
คณะวิศวกรรมศาสตร์

สอบกลางภาค: ภาคการศึกษาที่ 2 **ปีการศึกษา**: 2546

วันที่สอบ: 28 ธันวาคม 2546 **เวลาสอบ**: 9.00 – 12.00 น.

รหัสวิชา: 240-204 **ห้องสอบ**: 1207 1405

ชื่อวิชา: Data Structure and Computer Programming Techniques

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ

ไม่อนุญาต: หนังสือ, เอกสารใดๆ และเครื่องคิดเลข

เวลา: 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ข้อสอบมี 7 หน้า (รวมใบปะหน้า) 4 ข้อ **ให้ทำทุกข้อ** คะแนนรวม 100 คะแนน
- อย่าลืมเขียนชื่อ นามสกุลและรหัสประจำตัวนักศึกษา ลงทั้งในตัวข้อสอบและสมุดคำตอบ
- คำตอบทั้งหมดจะต้องเขียนลงใน**สมุดคำตอบ**
- ในการเขียนคำตอบ ให้ขึ้นหน้าใหม่เมื่อเริ่มตอบคำถามข้อใหม่(ยกเว้นข้อย่อยในข้อ 2.2)
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- การเขียนโปรแกรม ให้ใช้ภาษา C++
- อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ

#1 (20 คะแนน)	#2 (30 คะแนน)	#3 (20 คะแนน)	#4 (30 คะแนน)	Total (100 คะแนน)

คำถามที่ 1*(20 คะแนน)*

1.1 (10 คะแนน)

จงเขียนฟังก์ชันชื่อ `copy_str` โดยให้ใช้ พอยน์เตอร์ **(ห้ามใช้อาร์เรย์และห้ามใช้ฟังก์ชันstrcpy ใน standard library)** ซึ่งมี `function prototype` ดังนี้

```
void copy_str(char *dist, char *source);
```

ฟังก์ชัน `copy_str` จะทำให้สตริง `dist` มีข้อความเหมือนกับสตริง `source` ตัวอย่างการเรียกใช้ฟังก์ชัน `copy_str` จาก `main` เช่น

```
char str[20];
copy_str(str,"Computer");
cout>>str;     // ข้อความที่พิมพ์ออกมาคือ Computer
```

1.2(10 คะแนน)

จงเขียนฟังก์ชัน `search_number` เพื่อให้ทำงานได้ตามที่กำหนด โดยมีข้อแม้ว่าจะต้องเขียนเป็นฟังก์ชันที่มีเรียกตัวเองซ้ำ (Recursive function) **ห้ามมีการใช้คำสั่งในการวนรอบ (`while, for, do-while`)**

ฟังก์ชัน `how_many_equal_number` ใช้สำหรับหาจำนวนของเลขที่ซ้ำกันในอาร์เรย์ โปรโตไทป์ของฟังก์ชันเป็นดังนี้

```
int how_many_equal_number(const int A[],
                          int N, //ขนาดของอาร์เรย์(10)
                          int desired_value);
```

ตัวอย่างเช่น

ถ้าอาร์เรย์ `A` เก็บค่า `1,2,4,4,5,6,7,8,9` และ`12` เมื่อเรียกฟังก์ชันและผ่านค่าพารามิเตอร์ `how_many_equal_number(A,10,4)` จะได้ค่า 2 ส่งกลับมาจากฟังก์ชัน เนื่องจากในอาร์เรย์`A` มีเลข 4 อยู่จำนวน 2 ตัว

คำถามที่ 2*(30 คะแนน)*

กำหนดให้โครงสร้างของข้อมูลชื่อ Employee เพื่อใช้เก็บข้อมูลของพนักงาน ดังนี้

```
class Employee{

private:
      char*       first_name; //ใช้เก็บชื่อของพนักงาน
      char*       last_name;  //ใช้เก็บนามสกุลของพนักงาน
      int         employee_id;//ใช้เก็บเลขประจำตัวของพนักงาน
      double      salary;     //ใช้เก็บอัตราเงินเดือนของพนักงาน
public:
      void increase_salary(double percent_increase);


};
```

2.1(5 คะแนน)

จาก class Employee ข้างต้น จงเขียนฟังก์ชัน increase_salary เพื่อใช้ในการเพิ่มอัตราเงินเดือน (salary) ให้พนักงาน (Employee) โดยกำหนดให้ฟังก์ชัน increase_salary มีพารามิเตอร์ 1 ตัว คือ เปอร์เซนต์เงินเดือนขึ้น (percent_increase) ฟังก์ชัน increase_salary มี function prototype ดังนี้

```
void increase_salary(double percent_increase);
```

2.2(25 คะแนน)

จงเขียน `main function` สำหรับไฟล์ชื่อ `increase_employee_salary.cpp` เพื่อให้ทำงานดังนี้

2.2.1*(5 คะแนน)*

อ่านค่า `argument` จาก `command line` และตรวจสอบว่าได้รับ `argument` จาก `command line (command line argument)` ถูกต้องหรือไม่ ถ้าผลการตรวจสอบจำนวนแล้วพบว่าไม่ถูกต้อง ให้ `print error message` และหยุดการทำงานของโปรแกรม รายละเอียดของ `argument` มีดังนี้

`argument` ที่ 1 คือ	ชื่อของไฟล์ที่เก็บข้อมูลของพนักงาน `current_emp_data.txt`
`argument` ที่ 2 คือ	ชื่อของไฟล์ที่เก็บข้อมูลของพนักงานหลังจากที่ได้รับการขึ้นเงินเดือน `updated_emp_data.txt`
`argument` ที่ 3 คือ	เปอร์เซนต์เงินเดือนขึ้น

2.2.2*(10 คะแนน)*

อ่านข้อมูลของพนักงานจากไฟล์ `current_emp_data.txt`

(**ข้อแนะนำ :** น.ศ.สามารถสร้าง `class method` สำหรับ `class employee` สำหรับการอ่านข้อมูลของพนักงานจากไฟล์และรูปแบบการจัดเก็บข้อมูลมีแสดงอยู่หน้าที่ 5)

2.2.3*(10 คะแนน)*

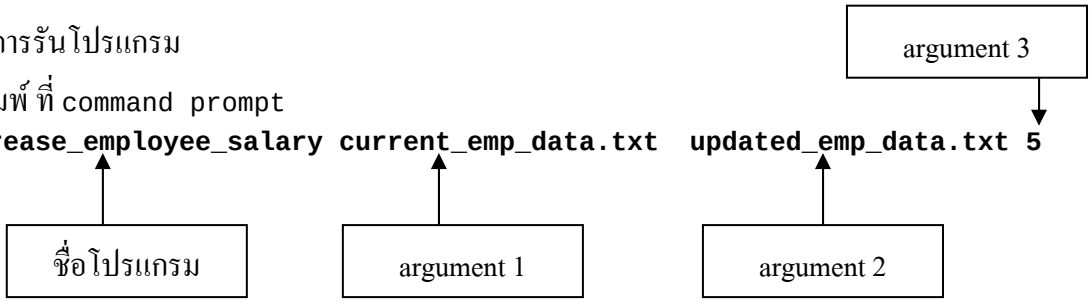
เรียกใช้ ฟังก์ชัน `increase_salary` (ที่ทำไว้ในข้อ2.1) เพื่อเพิ่มเงินเดือนพนักงานทุกคน 5 เปอร์เซนต์และให้ บันทึกข้อมูลใหม่ลงในไฟล์ชื่อ `updated_emp_data.txt`

(**ข้อแนะนำ :** น.ศ.สามารถสร้าง `class method` สำหรับ `class employee` สำหรับการเขียนข้อมูลของพนักงานลงไฟล์)

ตัวอย่างการรันโปรแกรม

User พิมพ์ ที่ `command prompt`

```
~> increase_employee_salary current_emp_data.txt  updated_emp_data.txt 5
```



ตัวอย่างข้อมูลในไฟล์ current_emp_data.txt

```
100                     ->employee_id
Pongpol                 ->first_name
Adireksarn              ->last_name
10000.00                ->salary
200
Adisai
Bodharamik
12000.00
300
Anurak
Chureemas
50000.00
........
........
........
........
```

ตัวอย่างข้อมูลในไฟล์updated_emp_data.txt หลังจากการรันโปรแกรมincrease_emp_salary:

```
100
Pongpol
Adireksarn
10500.00          ->new salary=old salary + 5% of the old salary
200
Adisai
Bodharamik
12600.00          ->new salary=old salary + 5% of the old salary
300
Anurak
Chureemas
52500.00          ->new salary=old salary + 5% of the old salary
........
........
........
........
```

คำถามที่ 3*(20 คะแนน)*

จงเขียนโค๊ด(implementation)ของ ฟังก์ชัน

- insertAt (6คะแนน)
- seqSearch (7 คะแนน)
- removeAt (6 คะแนน)

ตาม function prototype และให้ทำงานได้ตามข้อกำหนดดังนี้

```
class arrayListType{

public:

      void insertAt(int location, int insertItem);
            //Inserts an item in the list at the specified location.
            //The item to be inserted and the location are passed
            //as parameters to the function
            //If the list is full or the location is out of range,
            //an appropriate message is displayed.

      int seqSearch(int item);
            //Searches the list for a given item. If the item is
            //found, returns the location in the array where the
            //item is found; otherwise, returns -1

      void removeAt(int location);
            //Removes the item from the list at the specified
            //position. The location of the item to be removed is
            //passed as a parameter to this function.
            //If the list is empty or the location is out of range,
            //an appropriate message is displayed.



private:

      int *list;        //array to hold the list elements
      int length;       //stores the length of the list
      int maxSize;      //stores the maximum size of the list
};
```

คำถามที่ 4*(30 คะแนน)*

จงเขียนฟังก์ชัน intersection เพื่อให้ทำงานได้ตามที่กำหนด โดยมี function prototype และข้อกำหนดดังนี้

```
  node_type* intersection(node_type* L1, node_type* L2)

//Precondition: L1, L2 are linked lists both sorted in ASCENDING
//              order of their integer elements. There are no
//              duplicates in a list.

//Post condition: returns an entirely new list (L3 as shown in the
//                example)which is the reverse intersection of the
//                elements of the two input lists(L1&L2).


Typedef struct
{
      int         info;
      nodeType* link;
} node_type;
```

(**ข้อแนะนำ:** L3 ได้จากการนำ element ที่มีทั้งใน L1 และ L2 มาเรียงจากมากไปหาน้อย ให้ดูตัวอย่างข้างล่างประกอบ)

ตัวอย่าง:

Input: L1->[2]->**[9]**->**[17]**	เรียงจากน้อยไปมาก (ASCENDING)
Input: L2->[3]->**[9]**->[12]->**[17]**->[20]	เรียงจากน้อยไปมาก (ASCENDING)
output:L3->[17]->[9]	**เรียงจากมากไปน้อย**

//----------------------------------//