

**ข้อ 1** จงเขียนโปรแกรมให้ฟังก์ชัน main รับจำนวนเต็มจากผู้ใช้งานจำนวน 4 ค่า เก็บในอาร์เรย์ a[4] จากนั้นตรวจสอบค่าสมาชิกแต่ละตัวในอาร์เรย์ a ถ้าหากค่าสมาชิกตัวใดมากกว่าหรือเท่ากับศูนย์ ให้เรียกฟังก์ชัน add2 เพื่อบวก 2 ให้กับค่า นั้น ๆ และถ้าค่าสมาชิกใดต่ำกว่าศูนย์ ให้เรียกฟังก์ชัน minus2 เพื่อนำค่านั้นลบออกด้วย 2 จากนั้นฟังก์ชัน main จะแสดงค่าสมาชิกแต่ละค่าในอาร์เรย์ a ออกทางหน้าจอ ( 4 คะแนน )

กำหนดให้ ฟังก์ชันโปรโตไทป์ดังนี้

```
int add2 (int x);
int minus2 (int x);
```

```
#include<stdio.h>
int add2(int x);
int minus2(int x);
int main(){
    int i,a[4];
    for(i=0;i<4;i++){
        printf("Enter a[%d] : ",i);
        scanf("%d",&a[i]);
    }
    for(i=0;i<4;i++){
        if(a[i]>=0)
            a[i] = add2(a[i]);
        else
            a[i] = minus2(a[i]);
    }
    for(i=0;i<4;i++){
        printf("a[%d] = %d\n",i,a[i]);
    }
    return 0;
}

int add2(int x){
    return x+2;
}
int minus2(int x){
    return x-2;
}
```

#### ตัวอย่างการรันโปรแกรม

(กำหนดให้ค่าตัวเอียงหน้า เป็นค่าที่

ป้อนโดยผู้ใช้)

```
Enter a[0] : 2
Enter a[1] : -5
Enter a[2] : 10
Enter a[3] : -8
a[0] = 4
a[1] = -7
a[2] = 12
a[3] = -10
```

ข้อ 2 จงเขียนแกรมเรียงค่าจำนวนตัวเลข ที่มีการทำงานดังนี้ ( 6 คะแนน )

1. อาร์เรย์ A และ B มีจำนวนสมาชิกสูงสุด 20 ค่า
2. ฟังก์ชัน main รับจำนวนสมาชิกของอาร์เรย์ A จากผู้ใช้ แล้วเก็บในตัวแปร n
3. ฟังก์ชัน main รับค่าสมาชิกของอาร์เรย์ A ตัวที่ 1,2,...,n
4. ฟังก์ชัน main เรียกฟังก์ชัน arrange เพื่อเรียงค่าสมาชิกของอาร์เรย์ A ในอาร์เรย์ B โดยเริ่มจากสมาชิกตัวที่ n , n-1 , n-2 , ... , 1
5. ฟังก์ชัน main แสดงค่าสมาชิกในอาร์เรย์ B ทั้ง n ค่า ออกทางหน้าจอ

\*\*\*ให้กำหนด function prototype ของฟังก์ชัน arrange เอง

```
#include<stdio.h>
void arrange(int A[],int B[],int n);
int main(){
    int A[20],B[20],n = 20,i;
    printf("Enter n : ");
    scanf ("%d",&n);
    for(i=0;i<n;i++){
        printf("Emter A[%d] : ",i+1);
        scanf ("%d",&A[i]);
    }
    arrange(A,B,n);
    for(i=0;i<n;i++){
        printf("B[%d] = %d\n",i+1,B[i]);
    }
    return 0;
}

void arrange(int A[],int B[],int n){
    int i;
    for(i=0;i<n;i++){
        B[n-1-i] = A[i];
    }
}
```

ตัวอย่างการรันโปรแกรม

(กำหนดให้ค่าตัวเอียงหน้า เป็นค่าที่ป้อนโดยผู้ใช้)

```
Enter n : 4
Enter A0 : 12
Enter A1 : 11
Enter A2 : 10
Enter A3 : 13
B0 = 13
B1 = 10
B2 = 11
B3 = 12
```

ข้อ 3. กำหนดให้อาร์เรย์ name[20][30] ในฟังก์ชัน main ใช้เก็บชื่อและนามสกุลนักเรียน 20 คน ในห้องเรียน จงเขียนโปรแกรมสำหรับนับจำนวนอักษรที่ผู้ใช้ต้องการหาในชื่อและนามสกุลของนักเรียนทั้งหมด 20 คน

โดยโปรแกรมรับชื่อนามสกุลของนักเรียน และตัวอักษรที่ต้องการหาในฟังก์ชัน main แล้วเรียกใช้ฟังก์ชัน count เพื่อนับจำนวนอักษรที่กำหนด แล้วทำการแสดงจำนวนตัวอักษรที่นับได้ในแต่ละชื่อในฟังก์ชัน main กำหนดให้ count มีฟังก์ชันโปรโตไทป์ดังนี้

```
void count(char a[20][30] , char b , int n[20]); ( 10 คะแนน : 20 นาที)
```

โดยที่ a เป็นลิสต์รายชื่อนักเรียน , b คือตัวอักษรที่นับ , n คือจำนวนที่นับได้ในแต่ละชื่อ

```
#include<stdio.h>
#define N 20
void count(char a[20][30] , char b , int n[20]);
int main(){
    char a[20][30] , b;
    int n[20]={0};
    int i;
    for(i=0;i<N;i++){
        printf("Enter name %d: ",i+1);
        gets(a[i]);
    }
    printf("Find alphabet: ");
    scanf("%c",&b);
    count(a,b,n);
    for(i=0;i<N;i++){
        printf("%c in name %d = %d letters\n",b,i,n[i]);
    }
}

void count(char a[20][30] , char b , int n[20]){
    int i,j;
    for(i=0;i<N;i++){
        for(j=0;j<strlen(a[i]);j++){
            if(a[i][j]==b){
                n[i]++;
            }
        }
    }
}
```

#### ตัวอย่างการรันโปรแกรม

(กำหนดให้ค่าตัวเอียงหน้า เป็นค่าที่ป้อนโดยผู้ใช้)

```
Enter name 1: John Smith
Enter name 2: Malee Oscar
. . .
Enter name 20: Lindsey Lohan
Find alphabet: a
a in name 1 = 0 letters
a in name 2 = 2 letters
. . .
a in name 20 = 1 letters
```

## ตอนที่ 2 สตรักเจอร์ Structuer ( 30 คะแนน / 60 นาที)

ข้อ 1. จากสตรักเจอร์และตัวแปรที่กำหนดให้ จงเขียนส่วนของโปรแกรมเพื่อให้ทำงานตามคำสั่งในข้อ 1.1 – 1.6 (16 คะแนน)

**Subject** เป็นตรักเจอร์ สำหรับเก็บข้อมูลรายวิชา ประกอบด้วย รหัสวิชา (code) ชื่อวิชา (name) หน่วยกิต (credit) และเกรด (grade) ค่าเกรดเป็นค่า 0.0 – 4.0 เช่น เกรด 1.0 (D) 1.5 (D+) เป็นต้น

```
typedef struct{
    char code[10];
    char name[50];
    int credit;
    float grade;
}Subject;
Subject s1; float gpoints;
```

1.1 ประกาศตัวแปรชื่อ s2 สำหรับเก็บข้อมูลรายวิชา 240-101 intro to computer Programming 3 หน่วยกิต และได้เกรด 3.5 (B+) (กำหนดค่าเริ่มต้นของตัวแปร) (2 คะแนน)

```
Subject s2 = {"240-101","intro to computer Programming",3,3.5};
```

1.2 เขียนคำสั่งรับค่าต่าง ๆ ของรายวิชาสำหรับตัวแปร s1 (2 คะแนน)

```
gets(s1.code);
gets(s1.name);
scanf("%d",&s1.credit);
scanf("%f",&s1.grade);
```

1.3 เขียนคำสั่ง คำนวณค่า grade points คือผลคูณของหน่วยกิตและค่าเกรดของแต่ละวิชา ให้หาผลรวม grade points ของวิชา s1 และ s2 และเก็บผลรวมในตัวแปร gpoints (2 คะแนน)

```
gpoints = s1.credit*s1.grade + s2.credit*s2.grade;
```

1.4 เขียนคำสั่งแสดงค่าเกรดเฉลี่ยของวิชา s1 และ s2 (สามารถใช้ค่า gpoints จากข้อ 1.3 ได้) (2 คะแนน)

```
float avg;
avg = gpoints / (s1.credit+s2.credit);
```

1.5 เขียนคำสั่ง แสดงข้อมูลรายละเอียดของรายวิชา ตัวแปร s1 (2 คะแนน)

```
printf("%s",s1.code);
printf("%s",s1.name);
printf("%d",s1.credit);
printf("%f",s1.grade);
```

1.6 มีการประกาศอาร์เรย์สำหรับข้อมูลรายวิชา จำนวน 10 วิชา และมีการรับข้อมูลเก็บในอาร์เรย์ดังส่วนของโปรแกรมต่อไปนี้

```
Subject arrS[10];    // ข้อมูลรายวิชา

int n; ecredits = 0 ;

for(n=0; n<10 ;n++)

{    printf("Subject  #d: ",n+1);

    gets(arrS[n].code);

    gets(arrS[n].name);

    scanf("%d", &arrS[n].credit);

    scanf("%d", &arrS[n].grade);

}
```

หน่วยกิตที่ได้ (Earned credit) คือผลรวมของรายวิชาที่เรียนผ่าน (เกรด 1.0 ขึ้นไป ) จงเขียนส่วนของโปรแกรม (ไม่ต้องเขียนทั้งหมด) เพื่อคำนวณหน่วยกิตทั้งหมดที่ได้จากรายวิชา arrS และแสดงผลลัพธ์ (6 คะแนน)

```
for(n=0;n<10;n++){
.....
    if(arrS[n].grade >= 1){
.....
        ecredits += arrS[n].credit;
.....
    }
.....
}
.....
printf("%d",ecredits);
.....
.....
.....
```

**ข้อ 2.** จงเขียนโปรแกรม สำหรับคำนวณค่าจำนวนเชิงซ้อน (complex number) โดยโปรแกรมจะรับค่าจำนวนเชิงซ้อน 5 จำนวน แล้วค่าสัมบูรณ์ของจำนวนเชิงซ้อนเหล่านั้น พร้อมทั้งหาค่าผลรวมของจำนวนเชิงซ้อนทั้งหมด แล้วแสดงผลลัพธ์

โปรแกรมมีการประกาศสตรักเตอร์ Complex สำหรับเก็บค่าจำนวนเชิงซ้อน และมีค่าฟังก์ชัน cconst สำหรับสร้างจำนวนเชิงซ้อนจากจำนวนเต็มส่วนจริงและส่วนจินตภาพ และมีฟังก์ชัน cabsolute สำหรับคำนวณค่าสัมบูรณ์ ของจำนวนเชิงซ้อนจำนวนหนึ่ง

สำหรับจำนวนเชิงซ้อน  $z$  ,  $z = a + bi$  ค่าสัมบูรณ์  $|z|$  หรือ absolute value ของ  $z$  ซึ่งเป็นจำนวนจริง

สามารถคำนวณได้จากสูตร  $|z| = \sqrt{a^2 + b^2}$

นักศึกษาเติมโปรแกรมให้สมบูรณ์ (14 คะแนน)

```
#include<stdio.h>
#include<math.h>
typedef struct{
    int real;
    int imag;
}Complex;
Complex cconst(int re ,int im);
double cabsolute (Complex z);
int main(){
    int i,j,re,im; double a;
    Complex z[5], sum;
    printf("Enter 5 complex numbers \n");

    for(i= 0 ; i<5 ; i++){
        scanf("%d%d",&re,&im);

        z[i] = cconst( re , im );
    }
    sum = cconst(0,0);
    printf("Absolute value of complex numbers\n");
```

ตัวอย่างการทำงานของโปรแกรม

Enter 5 complex numbers

1 2

-2 1

3 4

-3 4

5 6

Absolute value of complex numbers

|1+2i| = 2.236

|-2+1i| = 2.236

|3+4i| = 5.000

|-3+4i| = 5.000

|5+6i| = 7.810

Sum = 4 + 17i

```

for(i= 0 ; i<5 ; i++) {

    a = cabsolute(z[i]);
    printf("|%d+%di| = %.4lf\n",z[i].real,z[i].imag,a);

    sum.real += z[i].real;

    sum.imag += z[i].imag;

}

printf("\nSum = %d + %di",sum.real,sum.imag);
return 0;
}

// นิยามฟังก์ชัน cconst (2 คะแนน)
Complex cconst(int re ,int im)
{
    Complex x;
    x.real = re;
    x.imag = im;
    return x;
}

// นิยามฟังก์ชัน cabsolute (2 คะแนน)
double cabsolute (Complex z)
{
    double a;
    a = sqrt(pow(z.real,2)+pow(z.imag,2));
    return a;
}

```

เพิ่มเติม.....

```
typedef struct{
    char code[10];
    int credit;
    float grade;
}Subject;
typedef struct{
    char stID[12]; // รหัสนักศึกษา
    char name[50]; // ชื่อ-สกุล
    int n; //จำนวนวิชาที่เรียน มีค่าไม่เกิน 10
    Subject s[10]; //ข้อมูลรายวิชา
}Student;
Student stA; // ประกาศตัวแปรเก็บข้อมูลนักศึกษา
```

ตัวอย่างการทำงานของโปรแกรม

```
Student ID : 6010110999
Name : Manee Meeta
Number of subjects : 3
Subject #1 Code : 240-101
Credits : 3
Grade : 2.5
Subject #1 Code : 890-102
Credits : 3
Grade : 0
Subject #1 Code : 322-113
Credits : 1
Grade : 1.5
-----
Mane Meeta's earned credits : 4
```

จงเขียนโปรแกรมที่สมบูรณ์ เพื่อรับข้อมูลนักศึกษาและวิชาที่เรียน แล้วให้คำนวณหน่วยกิตรวมที่ได้ (Earned credits) ซึ่งเป็นหน่วยกิตของวิชาที่เรียนผ่าน (เกรด 1.0 ขึ้นไป) เท่านั้น (10 คะแนน)

```
#include<stdio.h>
#include<string.h>
//นิยามสตรักเจอร์
int main(){
    int i;int SumCredit = 0;
    Student stA; //
    printf("Student ID : ");    scanf("%s",&stA.stID);
    fflush(stdin);
    printf("Name : ");
    gets(stA.name);
    printf("Number of subjects : ");
    scanf("%d",&stA.n);
    for(i=0;i<stA.n;i++){
        printf("Subject #%d Code : ",i+1);
        scanf("%s",&stA.s[i].code);
        printf("Credits : ");;
        scanf("%d",&stA.s[i].credit);
        printf("Grade : ");
        scanf("%f",&stA.s[i].grade);
    }
    for(i=0;i<stA.n;i++){
        if(stA.s[i].grade>1.0){
            SumCredit += stA.s[i].credit;
        }
    }
    printf("%s\'s earned credits : %d",stA.name,SumCredit);

    return 0;
}
```



ข้อ 4. จงเขียนโปรแกรมสำหรับคำนวณพื้นที่และเส้นรอบรูปของ รูปวงกลม โดยมีการประกาศสตรักเจอร์ Circle สำหรับเก็บข้อมูลของรูปวงกลม ดังนี้ รัศมี (radius) พื้นที่ (area) และความยาวเส้นรอบวง (circumference) ในโปรแกรมมีฟังก์ชัน createCircle ที่รับค่ารัศมี เป็นพารามิเตอร์ และฟังก์ชันคำนวณค่าพื้นที่ และความยาวรอบวงของรูปวงกลม และคืนค่าเป็นค่าข้อมูลเป็นชนิด Circle (10 คะแนน)

```
#include <stdio.h>
#define PI 3.142
typedef struct {
    float radius;
    float area;
    float circumference;
}Circle ;
Circle createCircle (float r);
int main(){
    Circle x;
    float r;
    printf("Enter radius : "); scanf("%f",&r);

    // กำหนดค่าให้กับตัวแปร x โดยเรียกใช้ฟังก์ชัน createCircle(2 คะแนน)

    x = createCircle(r);

    printf("Area of the circle = %f\n", x.area );
    printf("circumference of the circle =
%f\n", x.Circumference );
    return 0;
}
//จเขียนนิยามของฟังก์ชัน createCircle(6 คะแนน)

Circle createCircle (float r){
    Circle x;
    x.radius = r;
    x.area = PI*r*r;
    x.circumference = PI*2*r;
    return x;
}
```

### ตอนที่ 3 ( 20 คะแนน, 10%, 50 นาที)

จงเขียนโปรแกรมบัตรเครดิต โดยกำหนดให้โครงสร้างข้อมูลดังนี้

```
typedef struct
{
    int mm,yyyy; // month and year
}date;
typedef struct
{
    char name[21]; // Name of article
    float price; // Price of article
}article;
typedef struct
{
    char num[20]; // Card Number - 1234 1234 1234 1234

    int csc; // Card Security Code (CSC) - 123

    char name[80]; // Card Holder - Somchai Prapat

    date expire; // Expired Date - mm/yyyy

    float amount; // Amount - 50,000

    article art[100]; // List of articles

    int artnum; // Number of articles

}card;
```

ข้อ 1. จงเขียนฟังก์ชัน validate สำหรับการตรวจสอบบัตรเครดิตว่าหมดอายุหรือไม่ โดยฟังก์ชันจะต้องส่งค่า 0 กลับมาในกรณีที่บัตรหมดอายุ และ ส่งค่า 1 กลับมา ในกรณีที่บัตรใช้งานได้ (8 คะแนน) ตัวอย่างการเรียกใช้

```
card c;    date d = {11, 2560};    int v;
v = validate(c,d);
```

จงเขียนนิยามของฟังก์ชัน int validate(card c, date d);

```
int validate(card c, date d){
    if(c.expire.mm==d.mm && c.expire.yyyy==d.yyyy){
        return 0;
    }
    // C = 2550 , d = 2560 ปีปัจจุบันมากกว่า ปีในบัตร ( d คือปีปัจจุบัน , C คือบัตร )
    if(c.expire.yyyy < d.yyyy){
        return 0;
    }
    else if(c.expire.yyyy > d.yyyy){// C > 2560 , d = 2550
        return 1;
    }
    else{ // C = 2560 , d = 2560
        if(c.expire.mm < d.mm){ // c.mm = 5 , d.mm = 10
            return 0;
        }
        else{// c.mm = 10 , d.mm = 5
            return 1;
        }
    }
}
```

ข้อ 2. จงเขียนฟังก์ชัน buy สำหรับการซื้อสินค้าด้วยบัตรเครดิต โดยฟังก์ชันจะรับค่า บัตร เดือน/ปี ชื่อสินค้า และ ราคา และจะต้องดำเนินการต่อไปนี้

- ตรวจสอบว่าบัตรใช้งานได้หรือไม่ / เรียกใช้ฟังก์ชัน validate - ตรวจสอบว่ารหัส CSC ตรงกับบัตรหรือไม่ โดยให้รับค่า CSC จากผู้ใช้และเปรียบเทียบ - ตรวจสอบว่ามีวงเงิน (amount) เพียงพอหรือไม่

- หากตรวจสอบเป็นไปตามเกณฑ์ที่กำหนด ให้เพิ่มสินค้าไว้ในบัตร (art, artnum) พร้อมปรับลดวงเงิน ตัวอย่างการเรียกใช้ (12 คะแนน)

```
card c; date d = {11, 2560};  
C = buy (c, d, "Shoe", 1200);  
C = buy (c, d, "T-shirt", 750) ;
```

ตัวอย่างผลการทำงาน

```
Buying Shoe for 1200.00 baht  
Enter CSC of 1234 -1234 -1234 -1234 : 123  
Success: Available Credit 48800.00  
Buying T-shirt for 750.00 baht  
Enter CSC of 1234 -1234 -1234 -1234 : 123  
Success: Available Credit 48050.00
```

จงเขียนนิยามของฟังก์ชัน card buy (card c, date d, char name[], float b) ;

```
card buy(card c, date d, char name[], float b){  
    int CSC ;  
    if(validate(c,d)==1){  
        printf("Buying %s for %.2f baht \n",name,b);  
        printf("Enter CSC of 1234 -1234 -1234 -1234 :");  
        scanf("%d",&CSC);  
        if(c.csc == CSC){  
            if(c.amount>b){  
                c.artnum = 0;  
                strcpy(c.art[0].name,name);  
                c.art[0].price = b;  
                c.amount -= b;  
            }  
        }  
    }  
}
```

ข้อ 3. จงเขียนฟังก์ชันหลัก เพื่อเรียกใช้ฟังก์ชันต่างๆ และแสดงการทำงานให้ถูกต้อง