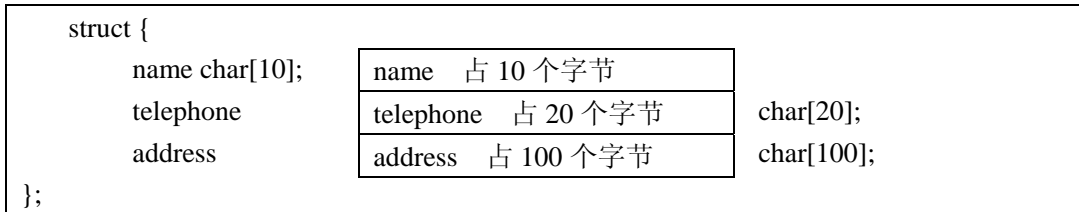


- A. typedef 语句用于定义新类型
- B. typedef 语句用于定义新变量
- C. typedef 语句用于给已定义类型取别名
- D. typedef 语句用于给已定义变量取别名

二. 编程题

1、设计一个通讯录的结构体类型，并画出该结构体变量在内存中的存储形式。



2、用结构体变量表示平面上的一个点（横坐标和纵坐标），输入两个点，求两点之间的距离。

```

#include <stdio.h>
#include <math.h>
void main()
{
    struct Point
    {
        float x;
        float y;
    }a,b;
    printf("输入点 a 的横坐标:\n");
    scanf("%f", &a.x);
    printf("输入点 a 的纵坐标:\n");
    scanf("%f", &a.y);
    printf("输入点 b 的横坐标:\n");
    scanf("%f", &b.x);
    printf("输入点 b 的纵坐标:\n");
    scanf("%f", &b.y);
    printf("a、b 两点之间的距离为:%f\n", sqrt((a.x - b.x)*(a.x - b.x) + (a.y - b.y)*(a.y - b.y)));
}
                
```

3、用结构体变量表示日期（年、月、日），任意输入两个日期，求它们之间相差的天数。

```

#include <iostream>
#include <stdio.h>
using namespace std;
#define DAY_OF_YEAR 365 //一年的天数，当为闰年时需要另外操作
//日期结构体
typedef struct date
{
    int year;
    int month;
}
                
```

```

int day;
int day_of_year; //该日为该年的第几天
}DATE, *DATE_P;
//函数声明
bool compear(DATE_P max, DATE_P min);
int titoal(DATE_P date);
bool leap_year(const int year);
int funtion(DATE_P max , DATE_P min);
//主函数
void main()
{
    DATE date_1, date_2; //输入的两个日期
    DATE_P max, min; //max 指向大日期, min 指向小日期
    int result;
    //数据输入
    printf("请输入日期 1 的数据项, 年月日间用空格隔开: \n");
    scanf("%d%d%d", &date_1.year, &date_1.month, &date_1.day);
    printf("请输入日期 2 的数据项, 年月日间用空格隔开: \n");
    scanf("%d%d%d", &date_2.year, &date_2.month, &date_2.day);
    //数据处理
    if( !compear(max=&date_1, min=&date_2) ) //比较日期大小,不需要交换
        result = funtion(max, min);
    else //需要交换
        result = funtion(min, max);
    //结果
    printf("两日期间相差的天数为: %d\n", result);
}
//比较两日期的大小
bool compear(DATE_P max, DATE_P min)
{
    //max 的日期小于 min 的日期, 日期对调
    if((max->year < min->year)) //max 的年份小于 min
        return 1;
    else if((max->year == min->year)) //max 和 min 的年份相同
    {
        if((max->month < min->month)) //max 的月份小于 min
            return 1;
        else if((max->month == min->month)) //max 和 min 的月份相同
            if((max->day < min->day)) //max 的日小于 min
                return 1;
    }
    //其他情况满足 max 的日期大于 (等于) min, 日期保持不变
    return 0;
}
    
```

```

//计算相差天数
int funtion(DATE_P max , DATE_P min)
{
    //年份相同,相差天数为大日期减小日期
    if(max->year == min->year)
    {
        //计算大日期
        max->day_of_year = titoal(max);           //计算日期函数
        if( leap_year(max->year) )                //如果为闰年
            if( (max->month==2 && max->day==29) || max->month>2)
                max->day_of_year ++;
        //计算小日期
        min->day_of_year = titoal(min);
        if( leap_year(min->year) )                //如果为闰年
            if( (min->month==2 && min->day==29) || min->month>2)
                min->day_of_year ++;
        //返回相差天数
        return max->day_of_year - min->day_of_year;
    }
    //年份不同,相差天数为 max、min 剩余天数和之间年的天数总和
    int max_year, min_year, other_year=0;
    //计算大日期
    max->day_of_year = titoal(max);
    if( leap_year(max->year) )                    //如果为闰年
        if( (max->month==2 && max->day==29) || max->month>2)
            max->day_of_year ++;
    max_year = max->day_of_year;
    //计算小日期
    min->day_of_year = titoal(min);
    if( leap_year(min->year) )                    //如果为闰年
        if( (min->month==2 && min->day==29) || min->month>2)
            min->day_of_year ++;
    min_year = DAY_OF_YEAR - min->day_of_year; //剩余的天数
    if( leap_year(min->year) )                    //如果为闰年天数加 1
        min_year ++;
    //计算之间的日期
    int year;
    for(year=min->year+1 ; year< max->year ; year++)
        other_year+=DAY_OF_YEAR+leap_year(year);//一年的天数, 并考虑是否为闰年
    //返回总天数
    return max_year + min_year + other_year;
}
//计算某日该年的第几天, 闰年也按平年处理
int titoal(DATE_P date)

```

```

{
    int result=0;           //最后结果
    int i;                 //标记月份
    for(i=1 ; i< date->month ; i++) //计算以过月份的天数
    {
        switch(i)
        {
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:result += 31;break;//大月 31 天，注：12 月不会出现，所以忽略不写
            case 4:
            case 6:
            case 9:
            case 11:result += 30;break;//小月 31 天
            default:result += 28;    //2 月 28 天
        }
    }
    return result += date->day;    //加上本月天数即为最后结果
}
//判断某年是否为闰年，若是返回 1，否则返回 0
bool leap_year(const int year)
{
    if( !(year%4) && year%100 || !(year%400) )
        return 1;
    return 0;
}

```

4、用结构体变量表示复数（实部和虚部），输入两个复数，求两复数之积。

```

#include <stdio.h>
struct com{
    int real;
    int im;
};
void main()
{
    struct com a,b,c;
    struct com cmult(struct com creal,struct com cim);
    int i,j;
    printf("输入两个复数的实部和虚部: ");
    scanf("%d%d%d%d",&a.real,&a.im,&b.real,&b.im);
    c=cmult(a,b);
}

```

```

        printf("乘积为: %d+%di",c.real,c.im);
    }
    struct com cmult(struct com creal,struct com cim)    /*求乘积的函数*/
    {
        struct com a;
        a.real=creal.real*cim.real-creal.im*cim.im;
        a.im=creal.real*cim.im+creal.im*cim.real;
        return(a);
    }
    /* 设复数为 a+bi 和 c+di , 那么 和为: (a+c)+(b+d)i, 乘积为 : (ac-bd)+(ad+bc)i 。
    */

```

- 5、有 20 个学生，每个学生的数据包括学号（num）、姓名（name）、性别（sex）、5 门课程成绩（score[5]）。要求编写一个程序，输入学生数据，计算并输出每个学生的总分和平均分。

```

#include <stdio.h>
#define N 20
void main()
{
    struct student
    {
        int num;
        char name[10];
        char sex[10];
        int score[5];
        int sum;
        float average;
    };
    struct student stu[N],s;
    int i,j;
    for(i=0;i<N;i++)
    {
        stu[i].sum=0;
        stu[i].average=0;
        printf("please scanf num name sex score(1) score(2) score(3) score(4) score(5)\n");
        scanf("%d%s%s",&stu[i].num,stu[i].name,stu[i].sex);
        for(j=0;j<5;j++)
        {
            scanf("%d",&stu[i].score[j]);
            stu[i].sum+=stu[i].score[j];
        }
        stu[i].average=stu[i].sum/5.0;
    }
    for(i=0;i<N;i++)

```

```

{
    printf("num name sex score(1) score(2) score(3) score(4) score(5) sum average\n");
    printf("%d %s %s ", stu[i].num, stu[i].name, stu[i].sex);
    for(j=0;j<5;j++)
    {
        printf("%d ",stu[i].score[j]);
    }
    printf("%d ", stu[i].sum);
    printf("%f ", stu[i].average);
    printf("\n");
}
}

```

- 6、箱子中有若干个红、黄、白 3 中颜色的小球，每次从中取出 2 个，求得到两种不同颜色的小球的可能情况，并输出每种颜色组合（使用枚举类型）。

```

#include <stdio.h>
#include <string.h>
void main()
{
    enum Color{Red, Yellow, White};
    enum Color i, j;
    for(i = Red; i <= White; i = (Color)(i + 1))
    {
        for(j = i; j <= White; j = (Color)(j + 1))
        {
            char s1[3], s2[3];
            switch(i)
            {
                case 0: strcpy(s1, "红"); break;
                case 1: strcpy(s1, "黄"); break;
                case 2: strcpy(s1, "白"); break;
            }
            switch(j)
            {
                case 0: strcpy(s2, "红"); break;
                case 1: strcpy(s2, "黄"); break;
                case 2: strcpy(s2, "白"); break;
            }
            printf("%s %s\n",s1,s2);
        }
    }
}

```

=====此后无内容=====