

1. 编写一个程序，用户输入 5 个整数，该程序通过调用函数，对 5 个整数按照从小到大的顺序排列并输出。

源代码:

```
#include <stdlib.h>
#include<stdio.h>
void print_sort(int *arr, int n);
int main()
{
    int a[5];
    int c;
    for(int i=0;i<5;i++){
        scanf("%d",&a[i]);
    }
    print_sort(a,5);
    return 0;
}

void print_sort(int *arr,int n){
    int b;
    int c=0;
    int i=0,j=0;
    for(i=0;i<n;i++){
        for(j=0;j<n-i;j++){
            if(*(arr+j)>*(arr+j+1)){
                b=*(arr+j+1);
                *(arr+j+1)=*(arr+j);
                *(arr+j)=b;
            }
        }
    }
    for(c;c<5;c++){
        printf("%d ",*(arr+c));
    }
}
```

输出:

```
5 4 1 2 3
1 2 3 4 5
```

2. 编写一个程序，用户输入一个日期和一个整数 n，该程序通过调用函数，输出该日期向后 n 天的日期，如果 n<0 即为向前的日期。需要考虑闰年。

源代码:

```
#include <stdio.h>
void find_date(int *p1,int *p2,int *p3,int x);
int main(void){
    int year,month,day;
    int n=0;
    scanf("%d %d %d %d",&year,&month,&day,&n);
    find_date(&year,&month,&day,n);
    printf("%02d-%02d-%02d\n",year,month,day);
    return 0;
}

void find_date(int *p1, int *p2, int *p3, int x){
    int a[12]={31,28,31,30,31,30,31,31,30,31,30,31};
    *p3=*p3+x;
    if(*p1%4==0){
        if (*p1 % 100 == 0){
            if (*p1 % 400 == 0){
                a[2] = 29;
            }
            a[2] = 28;
        }
        a[2] = 29;
    }
    else{
        a[2] = 28;
    }
    if (*p3 > a[*p2]){
        for (; *p3 > a[*p2];){
            *p3 = *p3 - a[*p2];
            *p2++;
            if (*p2 == 13){
                *p1++;
                *p2 = 1;
            }
        }
    }
    if (*p3 < 0){
        for (; *p3 < 0;){
            *p2 = *p2 - 1;
            *p3 = a[*p2] + *p3;
            if (*p2 == 0){
```

```
        *p2 = 12;  
        *p1--;  
    }  
}  
}
```

输出:

```
2000 2 28 3  
2000-02-02
```

3. 编写一个程序，用户输入 4 个学生的学号和他们在 5 门课程的成绩，该程序通过调用不同的函数，分别实现三个功能：(1) 计算第 n 门课程的成绩；(2) 找出有 n 门或 n 门以上课程不及格的学生，输出他们的学号和全部课程成绩及平均成绩；(3) 找出平均成绩大于等于 avg_score 或全部课程成绩大于等于 min_score 的学生，输出他们的学号和全部课程成绩及平均成绩。

源代码:

```
#include <stdio.h>
#define NUM_STUDENTS 4
#define NUM_CLASSES 5
#define ID_LENGTH 15
void avg_class(int *scores, int n);
void find_fail(char (*ids)[ID_LENGTH], int *scores,
int n);
void find_better(char *ids, int
(*scores)[NUM_CLASSES], int avg_score, int min_score);
int main(void){
    char ids[NUM_STUDENTS][ID_LENGTH];
    int scores[NUM_STUDENTS][NUM_CLASSES];
    int n=0;
    int avg_score=0,min_score=0;
    for(int i=0;i<NUM_STUDENTS;i++){
        scanf("%s",ids[i]);
        for (int j=0;j<NUM_CLASSES;j++){
            scanf("%d",&scores[i][j]);
        }
    }
    printf("\n 要计算平均成绩的课程序号: [0-
-%d]",NUM_CLASSES-1);
    scanf("%d",&n);
    avg_class(*scores,n);
    printf("\n 要统计不及格课程的数量: [1-
-%d]",NUM_CLASSES);
    scanf("%d",&n);
    find_fail(ids,*scores,n);
    printf("\n 不低于的平均成绩: ");
    scanf("%d",&avg_score);
    printf("不低于的全部成绩: ");
    scanf("%d",&min_score);
    find_better(*ids,scores,avg_score,min_score);
    return 0;
}

void avg_class(int *scores,int n){
    int i;
```

```

float sum=0,avg;
scores+=n;
for(i=0;i<4;i++){
    sum=sum+*scores;
    scores+=5;
}
avg=sum/4;
printf("序号%d 的课程平均成绩为%.2f\n",n,avg);
}

```

```

void find_fail(char (*ids)[ID_LENGTH],int *scores,int
n){
    int i,z=0,j,e;
    float sum=0,avg;
    for(j=0;j<4;j++){
        for(i=0;i<5;i++){
            if(*scores<60){
                z++;
            }
            scores++;
        }
        if(z>=n){
            ids+=j;
            printf("%s",*ids);
            scores-=5;
            printf("全部成绩:");
            for(e=0;e<5;e++){
                printf(" %d",*scores);
                sum+=*scores;
                scores++;
            }
            avg=sum/5;
            printf(" 平均成绩:%.2f",avg);
        }
        ids-=j;
        z=0;
    }
}

```

```

void find_better(char *ids,int
(*scores)[NUM_CLASSES],int avg_score,int min_score){
    int i,j,z=0,e,t,a;
    float sum=0,avg;
    for(i=0;i<4;i++){

```

```

        for(j=0;j<5;j++){
            if(*(scores+i)+j>=min_score){
                z++;
            }
            sum+=*(scores+i)+j);
        }
        avg=sum/5;
        if((z==5)|| (avg>=avg_score)){
            for(t=0;t<15;t++){
                printf("%c",*ids);
                ids++;
            }
            printf(" 全部成绩: ");
            for(e=0;e<5;e++){
                printf(" %d",*(scores+i)+e));
            }
            printf(" 平均成绩:%.2f\n",avg);
        }
        else{
            for(a=0;a<15;a++){
                ids++;
            }
        }
        sum=0;
        z=0;
    }
}

```

输出:

```

2019010501 61 63 65 67 69
2019010502 71 45 83 28 74
2019010503 83 92 95 78 89
2019010504 56 65 77 74 79
要计算平均成绩的课程序号: [0--4]3
序号 3 的课程平均成绩为 61.75
要统计不及格课程的数量: [1--5]2
2019010502 全部成绩: 71 45 83 28 74 平均成绩:60.20

```