

学号	2021144301	姓名	王大殿	实验日期	
实验名称	一维数组				
实验目的：					
1. 掌握一维数组的定义、赋值和输入输出的方法；					
2. 掌握与数组有关的算法（例如排序算法）。					
实验任务、步骤与结果：					
任务 1：数组插入数据					
题目描述					
有一个已排序的数组{2,5,7,10,14,17,20}，现输入一个整数，要求按原来排序的规律将它插入数组中。					
代码：					
<pre>#include<stdio.h> int main(){ int a[10]={2,5,7,10,14,17,20}; int n, i; scanf("%d",&n); for(i=0; i<7;i++){ if(n<=a[i]){ for(int j=7; j>i; j--){ a[j]=a[j-1]; //printf("%d\n",a[j]); } a[i]=n; //printf("%d\n",a[i]); break; }else if (n>a[6]) { a[7]=n; } } for(int k=0; k<8; k++){ printf("%d\t",a[k]); } printf("\n"); return 0; }</pre>					

运行结果:

```
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./数组插入
4
2      4      5      7      10     14     17     20
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./数组插入
31
2      5      7      10     14     17     20     31
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ █
```

任务 2: 数组逆序存储

题目描述:

将一个数组中的值按逆序重新存放。例如，原来顺序为 8，6，5，4，1。逆序后的顺序为 1，4，5，6，8。

代码:

```
#include<stdio.h>
int main(){
    int a[100];
    char ch;
    int len=0;
    do{
        scanf("%d%c",&a[len],&ch);
        len++;
    }while(ch!='\n');
    int a_mirror[len];
    for(int i=0; i<len; i++){
        a_mirror[i]=a[len-i-1];
        printf("%d ", a_mirror[i]);
    }
    printf("\n");
    return 0;
}
```

运行结果:

```
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ gcc 数组逆序输出.c -o 数组逆序输出
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./数组逆序输出
1 2 3 4 5 6
6 5 4 3 2 1
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./数组逆序输出
8 6 5 4 1
1 4 5 6 8
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ █
```

任务 3: 约瑟夫问题

题目描述

又称猴子选大王问题。 m 个猴子围成一圈，从 1 开始报数，报到 n 的猴子出局，剩下的猴子再从 1 开始报数，报到 n 者出局。最后剩下的猴子为大王。

编程输出最后留下的大王？

输入

1 行：

两个整数，猴子数 m 和报数 n

输出

大王的位置，如 King is a[1]=1

代码：

```
#include<stdio.h>
```

```
int main(){
```

```
    int m, n;
```

```
    printf("请输入猴子个数 m， 以及 n\n");
```

```
    scanf("%d%d",&m, &n);
```

```
    int len=m, i=1, k=0, count=0;
```

```
    //i 用来给循环计数，k 为数组下标，count 用于在循环中记录 0
```

```
    int a[len];
```

```
    for(int j=0; j<len; j++){
```

```
        a[j]=j+1;
```

```
    }
```

```
    while (len>1){
```

```
        k=(i+count-1)%m;
```

```
        if(a[k]!=0){
```

```
            if(i%n==0){
```

```
                a[k]=0;
```

```
                len--;
```

```
            }
```

```
            i++;
```

```
        }else{
```

```
            count++;
```

```
            //当 a[k]==0 时，count 加一，用于跳向下一位
```

```
        }
```

```
    }
```

```
    for(int i=0; i<m; i++){
```

```
        if(a[i]!=0){
```

```
            printf("King is a[%d]=%d\n",a[i],a[i]);
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

运行结果：

```

daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ gcc -g JosefProblem.c -o JosefProblem
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./JosefProblem
请输入猴子个数m, 以及n
8 5
King is a[3]=3
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./JosefProblem
请输入猴子个数m, 以及n
10 6
King is a[3]=3
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./JosefProblem
请输入猴子个数m, 以及n
20 2
King is a[9]=9
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ 

```

任务 4: 折半查找

题目描述

有 15 个数按由小到大顺序存放在一个数组中，输入一个数，要求用折半查找法找出该数是数组中第几个元素的值。如果该数不在数组中，则打印出 “None”。

代码:

```

#include<stdio.h>
int binarySearch(int arr[], int, int, int);
int main(){
    int len=15, index=0, n=0;
    printf("Input the number you want:");
    scanf("%d",&n);
    int a[len];
    for(int i=0; i<len; i++){
        a[i]=i+1;
    }
    index=binarySearch(a, 0, len, n);
    if(index!=-1){
        printf("The index of %d is %d.\n",n,index);
    }else{
        printf("None!\n");
    }
}
int binarySearch(int arr[], int low, int high, int x){
    int mid=(low+high)/2;
    if(x>arr[high-1] || x<arr[low]){
        return -1;
    }
    if(arr[mid]==x){
        return mid;
    }else if(x<arr[mid]){
        return binarySearch(arr, low, mid, x);
    }else if(x>arr[mid]){
        return binarySearch(arr, mid, high, x);
    }
}

```

运行结果

```
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./Binary_Search
Input the number you want:1
The index of 1 is 0.
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./Binary_Search
Input the number you want:5
The index of 5 is 4.
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./Binary_Search
Input the number you want:-1
None!
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./Binary_Search
Input the number you want:19
None!
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$
```

任务 5: 看到几座山

题目描述

众所周知，黄山是一片山（而不是一座山）。我们假设这些山排成了一排。每座山有各自的高度。现在游客们从最左边看山，有些山因为高度没有它左边的某座山高，就会被遮住，游客们就无法看到，也就是说对于一座山而言，只有当这座山比它左边所有的山都高的时候，游客们才能看到这座山。现在想请你告诉游客，他能看到几座山。

这个问题可以描述为找出数组中比前面所有元素都要大的元素个数。

代码：

```
#include<stdio.h>
int main(){
    int N=0;
    printf("Input the number of mountains: ");
    scanf("%d",&N);
    long int arr[N];
    char ch;
    int i=0, top=0, count=1;
    do{
        scanf("%ld%c",&arr[i],&ch);
        i++;
    }while(ch!='\n');
    for(int j=0; j<N-1; j++){
        if(arr[j]>top){
            top=arr[j];
        }
        if(arr[j+1]>top){
            count++;
        }
    }
    printf("You can see %d mountains.\n",count);
    return 0;
}
```

运行结果：

```
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ gcc 看到几座山.c -o 看到几座山
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./看到几座山
Input the number of mountains: 10
1 2 3 4 5 1 2 3 4 5
You can see 5 mountains.
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./看到几座山
Input the number of mountains: 5
5 4 3 2 1
You can see 1 mountains.
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$ ./看到几座山
Input the number of mountains: 5
3 4 1 8 5
You can see 3 mountains.
daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/实验/实验5$
```

发现问题与分析：

备注：

说明：

- 1、各项记录应完备详尽。
- 2、各栏目如果长度不够可以自行调整大小，但应注意排版的美观。
- 3、可将实现过程的代码、运行效果等内容以贴图方式放在相应栏目。