

```
#include <stdio.h>
#define M 1000
int main(void)
{
    int a[M],b[M],n,i,m,j1,j2;
    printf("请输入顺序表中数据元素的个数: ");
    scanf("%d",&n);
    j1=n-1;
    j2=0;
    for (i=0;i<n;i++)
    {
        scanf("%d",&m);
        a[i]=m;
    }
    printf("原顺序表为: ");
    for (i=0;i<n;i++)
    {
        printf("%d ",a[i]);
    }
    for (i=0;i<n;i++)
    {
        if (a[i]%2==1)
        {
            b[j2]=a[i];
            j2++;
        }
        else
        {
            b[j1]=a[i];
            j1--;
        }
    }
    printf("调整后的表为: ");
    for (i=0;i<n;i++)
    {
        printf("%d ",b[i]);
    }
}
```

```
DAG语言程序排序.exe
请输入顺序表中数据元素的个数: 9
15
23
28
10
7
567
20
80
66
原顺序表为: 15 23 28 10 7 567 20 80 66 调整后的表为: 15 23 7 567 66 80 20 10 28
Process exited after 28.13 seconds with return value 9
请按任意键继续. . .
```

时间复杂度为: $O(n)$

基本操作是第三个for循环中的赋值语句, 因为里面是if-else语句结构, 一次只执行一条赋值语句, 故一共执行n次, 故时间复杂度为 $O(n)$ 。

第二题

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node *p;
};
typedef struct node NODE;
typedef struct node *NODEp;
NODEp createlist(int l);
NODEp deletenode(NODEp list,NODEp link);
void display(NODEp list);
int main(void)
{
    NODEp List,a,b,list1;
    int u,max;
    printf("创建一个单链表, 链表长度为: ");
    scanf("%d",&u);
    List=createlist(u);
    printf("原链表为: ");
    display(List);
    max=(List->data);
    for(a=List;a!=NULL;a=a->p)
    {
        if (a->data>max)
        {
            max=a->data;
            b=a;
        }
    }
    printf("\n该链表的最大数据为:%d",b->data);
```

```

list1=deletenode(List,b);
printf("\n删除后的链表为: ");
display(list1);
}
NODEp createlist(int l)
{
    NODEp m,n,list=NULL;
    int i;
    for (i=0;i<l;i++)
    {
        m=(NODEp)malloc(sizeof(NODE));
        scanf("%d",&(m->data));
        m->p=NULL;
        if (list==NULL)
        {
            list=m;
            n=m;
        }
        else
        {
            n->p=m;
        }
        n=m;
    }
    return list;
}
NODEp deletenode(NODEp LIST,NODEp t)
{
    NODEp e;
    if (t==LIST)
    {
        LIST=LIST->p;
        free(t);
    }
    else
    {
        for(e=LIST;e->p!=t&&e->p!=NULL;e=e->p)
        {
            ;
        }
        if (e->p!=NULL)
        {
            e->p=t->p;
            free(t);
        }
    }
    return LIST;
}
void display(NODEp list)
{
    NODEp r;
    r=list;
    if (r==NULL)
    {
        printf("Error");
    }
}

```

```
}  
while (r)  
{  
    printf("%d->",r->data);  
    r=r->p;  
}  
}
```

```
DAC语言程序找最大数据.exe  
创建一个单链表，链表长度为：7  
15  
27  
89  
127  
56  
66  
99  
原链表为：15->27->89->127->56->66->99->  
该链表的最大数据为：127  
删除后的链表为：15->27->89->56->66->99->  
-----  
Process exited after 18.59 seconds with return value 0  
请按任意键继续. . .
```