



应用示例

1.试写出下面程序的运行结果。

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

```
#define n 10
```

```
void main( )
```

```
{  int a[ ]={12, 34, 2, 56, 78, 89, 8, 11, 22, 6};
```

```
    int b[n] , i , j=0, k=n;
```

```
    for (i=0; i<n;i++)                //语句1
```

```
        if (a[i]/10) b[j++]=a[i];      //语句2
```

```
        else b[--k]=a[i];              //语句3
```

```
    for (i=0; i<n;i++)
```

```
        printf("%4d",b[i]);
```

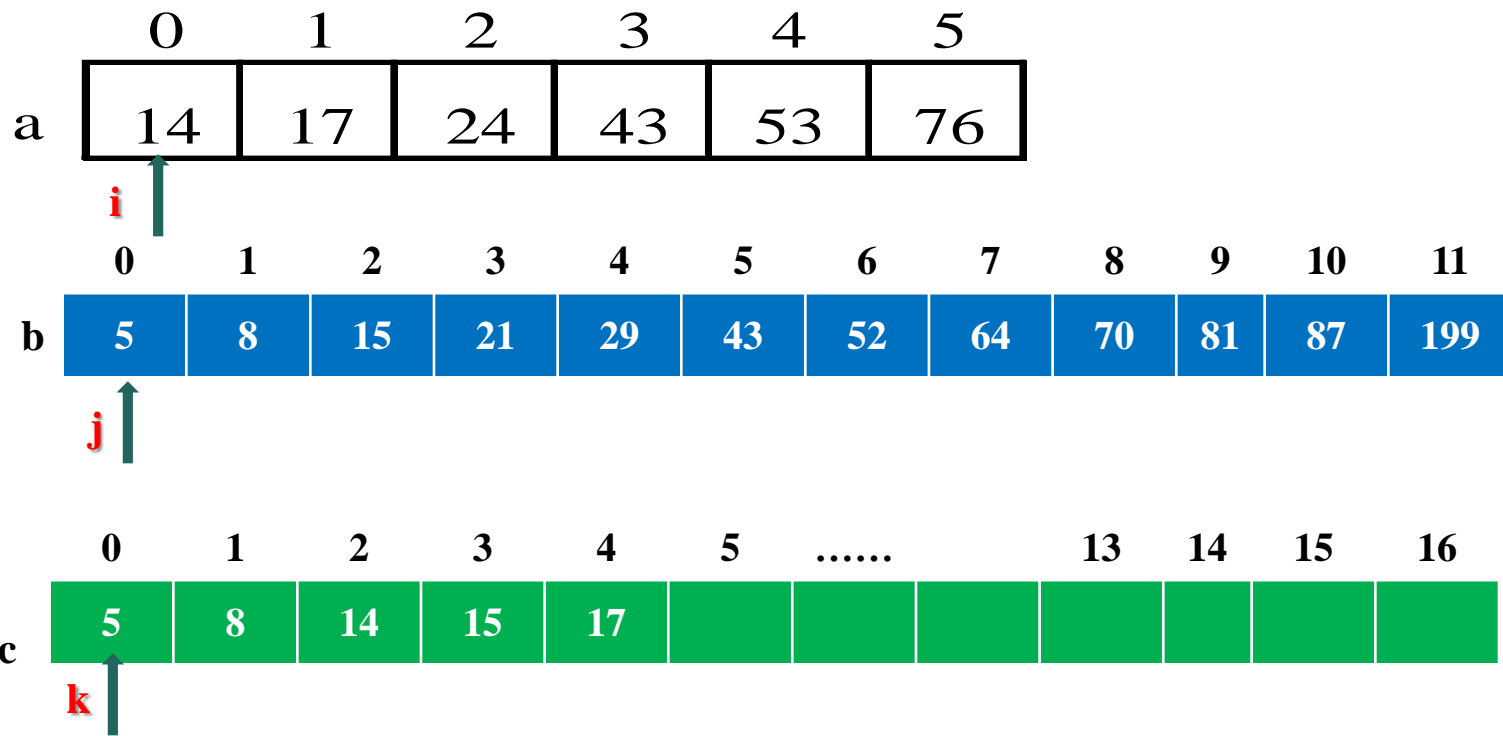
```
}
```

```
b[ ]={ 12, 34, 56, 78, 89, 11, 22, 6, 8, 2};
```



应用示例

2.设计一个函数，将两个顺序存储的严格递增的有序表**合并**成一个严格递增的有序表。





应用示例

2.设计一个函数，将两个顺序存储的严格递增的有序表**合并**成一个严格递增的有序表。

```
int merge(int a[ ],int m,int b[ ],int n,int c[ ])
    //参数m，n分别是数组a和b的长度
{ int i=0,j=0,k=0;
  while(i<m && j<n)
  { if (a[i]<b[j]) c[k++]=a[i++];
    else if (b[j]<a[i]) c[k++]=b[j++];
    else c[k++]=b[j++],i++;
  }
  while(i<m) c[k++]=a[i++]; //复制剩余数据段
  while(j<n) c[k++]=b[j++]; //复制剩余数据段
  return k; //返回值k是数组c的长度
}
```



3. 多项式求和

❖ $P = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n$

❖ $Q = b_0 + b_1x + b_2x^2 + b_3x^3 + \dots + b_mx^m$

❖ 求 $P+Q$

p	a₀	a₁	a₂	a₃	...	a_n
	0	1	2	3		n

Q	b₀	b₁	b₂	b₃	...	b_n	b_{n+1}	...	b_m
	0	1	2	3		n			m

P+Q	a₀ +b₀	a₁ +b₁	a₂ +b₂	a₃ +b₃	...	a_n +b_n	b_{n+1}	...	b_m
	0	1	2	3		n			m



稀疏多项式如何用计算机处理？

❖ 例： $P=8+10x^2+5x^{100}$

8	0	10	0	0	...	0	5
0	1	2	3				100

❖ 为了提高存储空间利用率，可采用压缩存储法

	系数	指数
0	8	0
1	10	2
2	5	100

a

```
typedef struct element
{
    int ceof;//定义系数
    int exp;//定义指数
}elem;
elem a[N];
```



稀疏多项式求和

❖ $P=8+10x^2+5x^{100}$

❖ $Q=2x^2+9x^{17}-5x^{100}+15x^{130}$

❖ 求P+Q

P	ceof	exp
	8	0
	10	2
	5	100

Q	ceof	exp
	2	2
	9	17
	-5	100
	15	130

P+Q	ceof	exp
	8	0
	12	2
	9	17
	15	130



稀疏多项式求和

压缩存储后为 $P[N]$, $Q[M]$, $P+Q$ 存储在新表 $C[N+M]$ 中

```
i=j=k=0;
while(i<N && j<M)
{
    if(P[i].exp<Q[j].exp)
    { C[k++]=P[i];
      i++;
    }
    else if(P[i].exp>Q[j].exp)
    { C[k++]=Q[j];
      j++;
    }
}
```

```
else
{
    if((P[i].ceof+Q[j].ceof)!=0)
    { C[k++].ceof=val;
      C[k++].exp=P[i].exp;
    }
    i++;j++;
}
}
最后将P或Q的剩余段加入新表中
```



稀疏多项式求和

- ❖ 如果在原表（P或Q）上进行相加运算，就需要进行插入（指数不等时）和删除（指数相等并系数相加为0时）运算，效率较低。
- ❖ 如何进行高效操作？