

# 动态数组

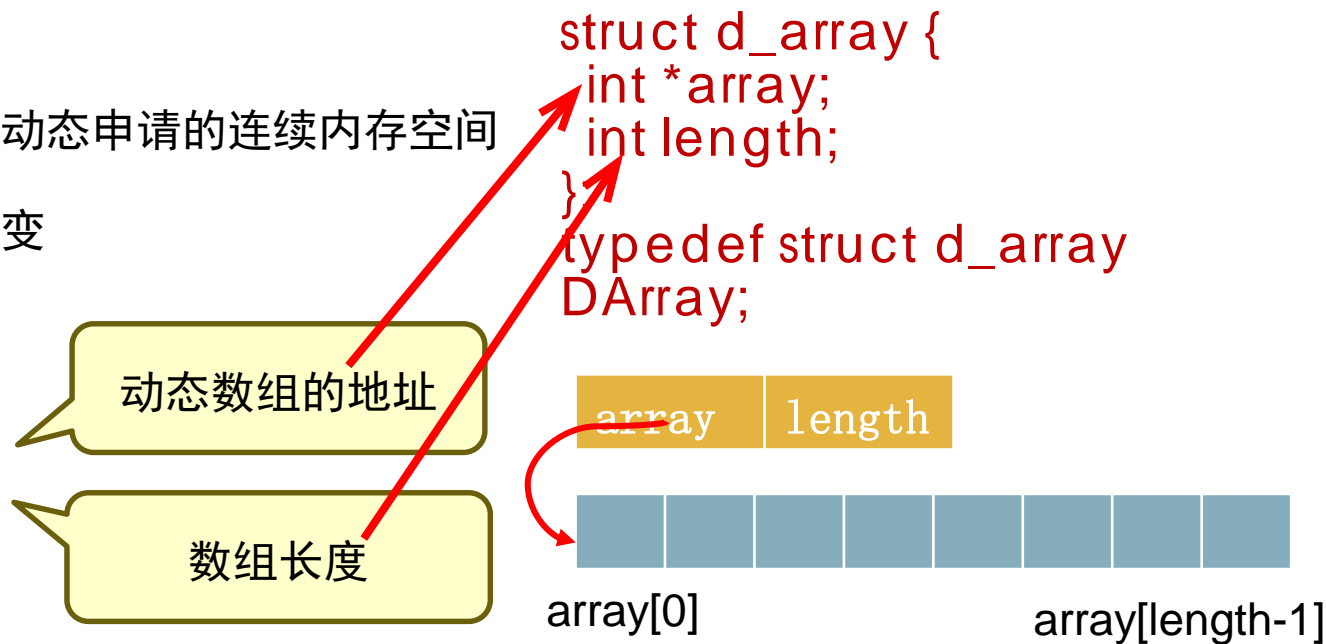
## \* 动态数组

特点	静态数组	动态数组
创建	定义	内存申请函数
释放	不需要	程序员
内存区域	不定	堆
生存期	不定	程序
数组长度	不变	可变

## 动态数组

### \* 动态数组

- \* 程序运行中动态申请的连续内存空间
- \* 长度动态可变

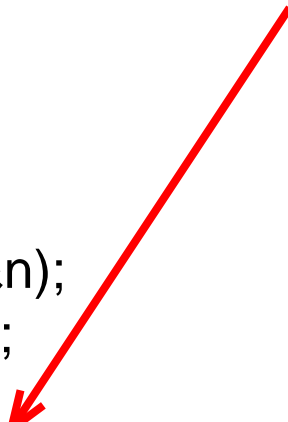


## 动态数组

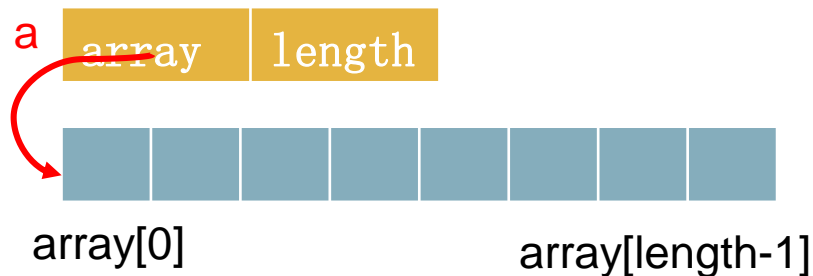
**/Example:** 创建一个长度为n的动态数组，读入数据后将其长度增长1倍，继续读入新的数据。输出长度增长前后的所有元素。

```
DArray create(int );  
void grow(DArray *, int);  
void print(DArray *);  
void release(DArray *);
```

```
int main()  
{   int n;  
    DArray a;  
    scanf("%d",&n);  
    a = create(n);  
    print(&a);  
    grow(&a, n*2);  
    print(&a);  
    release(&a);  
    return 0;  
}
```



## 动态数组



```
DArray create(int n)
```

```
{
```

```
    DArray a;
```

```
    int i;
```

```
    a.array = (int *)malloc(sizeof(int)*n);
```

```
    if(a.array == NULL)
```

```
    {
```

```
        printf("Allocation Error");
```

```
        exit(0);
```

```
    }
```

```
    else
```

```
    { a.length = n;
```

```
      for(i=0; i<a.length; i++)
```

```
        scanf("%d",&a.array[i]);
```

```
    }
```

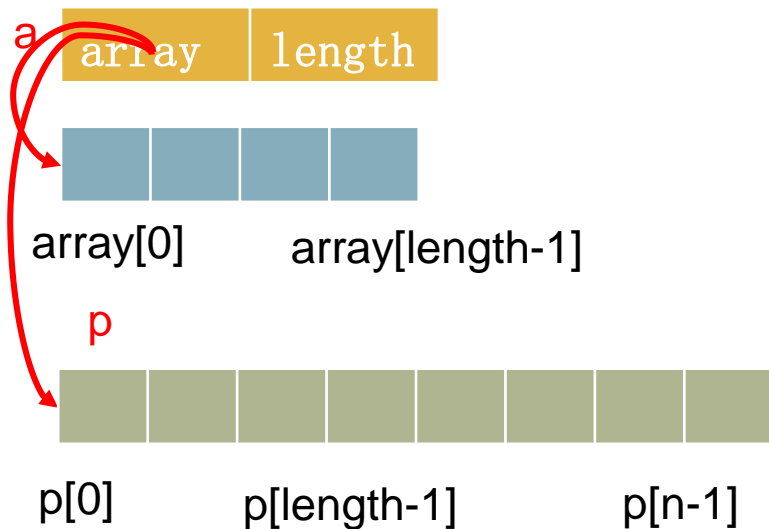
```
    return a;
```

```
}
```

授课：

若create传入的为结构体  
变量指针，该如何修改

## 动态数组



```

void grow(DArray *aPtr, int n)
{
    int *p;
    int i;
    p = (int *)malloc(sizeof(int)*n);
    if(p==NULL)
        return;
    else
    {
        for(i=0; i<aPtr->length; i++)
            p[i] = aPtr->array[i];
        for(i=aPtr->length; i<n; i++)
            scanf("%d",&p[i]);
        free(aPtr->array);
        aPtr->array = p;
        aPtr->length = n;
    }
}

```

`p = (int*)realloc(aPtr->array, sizeof(int) * n);`

若修改代码，改为：  
`malloc -> realloc`  
 问题：  
 1. 第1个for循环还需要吗  
 2. free还需要吗