

# 指 针

指针与数组



# 指向数组元素的指针

```
#include "stdio.h"

int main()
{   int i;

    char s[5]={'a','b','c','d','e'};

    char *p;

    p=&s[0];

    for(i=0;i<5;i++,p++)

        printf("%c",*p);

return 0;

}
```

数组元素在内存中是连续存放的

运行结果：

abcde

	内存单元	地址
s	'a'	2424368
	'b'	2424369
	'c'	2424370
	'd'	2424371
	'e'	2424372
	.....	
p	2424373	



```

#include "stdio.h"

int main()
{
    int i;

    char s[3][3]={'a','b','c','d','e','f','g','h','i'};

    char *p;

    p=&s[0][0];

    for(i=0;i<9;i++,p++)
        printf("%c",*p);

}

```

运行结果：

abcdefghi

数组s的逻辑结构

'a'	'b'	'c'
'd'	'e'	'f'
'g'	'h'	'h'

	内存单元	地址
s	'a'	2424368
	'b'	2424369
	'c'	2424370
	'd'	2424371
	'e'	2424372
	'f'	2424373
	'g'	2424374
	'h'	2424375
	'i'	2424376
	.....	.....
p	2424377	



# 指向数组的指针

```
#include "stdio.h"
```

```
int main()
```

```
{ int i;
```

```
int x[5]={ 9,4,6,3,7};
```

```
int (*p)[5]; //比较 int *p[5];
```

```
printf("数组的首地址 %d\n",x );
```

```
p=&x;
```

```
printf("输出指针值 %d\n",p);
```

```
printf("输出指针所指数组的地址 %d\n",*p);
```

```
printf("用指针变量获取数组元素:\n" );
```

```
for(i=0;i<5;i++) printf("%d ",(*p)[i]);
```

```
return 0;
```

```
}
```

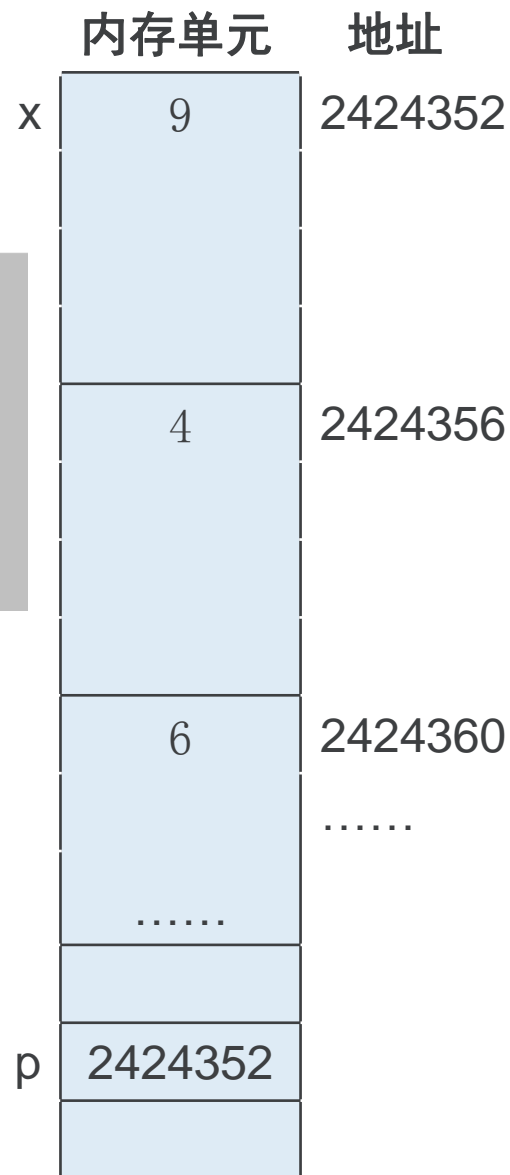
数组的首地址 2424352

输出指针值 2424352

输出指针所指数组的地址 2424352

用指针变量获取数组元素:

9 4 6 3 7



```
#include "stdio.h"

int main()
{   int i;
    char s[3][3]={'a','b','c','d','e','f','g','h','i'};
    char (*p)[3];
    printf("数组的首地址 %d\n",s );
    printf("    输出每一行首地址 %d %d %d\n",s[0],s[1],s[2]) ;
    p=s;
    printf("用指针输出每一行首地址 %d %d %d\n",p,p+1,p+2);
    printf("用指针输出第一行的数组元素:\n");
    for(i=0;i<3;i++)   printf("%c ",p[0][i]);
    printf("\n用指针输出第二行的数组元素:\n");

    p++;
    for(i=0;i<3;i++)   printf("%c ",(*p)[i]);
    return 0;
}
```

数组s的逻辑结构			
s[0]	'a'	'b'	'c'
s[1]	'd'	'e'	'f'
s[2]	'g'	'h'	'i'

	内存单元	地址
s	'a'	2424368
	'b'	2424369
	'c'	2424370
	'd'	2424371
	'e'	2424372
	'f'	2424373
	'g'	2424374
	'h'	2424375
	'i'	2424376
p	.....	.....
	2424371	

数组的首地址 2424368  
输出每一行首地址 2424368 2424371 2424374  
用指针输出每一行首地址 2424368 2424371 2424374  
用指针输出第一行的数组元素:  
a b c  
用指针输出第二行的数组元素:  
d e f

# THANKYOU

