

字符串

1

字符串概念

字符串 = 以 '\0' (空字符) 结尾的字符数组

```
char str[] = "hello";
```



在内存中实际是

'h' 'e' 'l' 'l' 'o' '\0'

'\0' 非常重要，字符串结束标志

设有数组定义: char str []="Tianjin\n"; 则数组 array 所占的空间为 B 。

- A. 10 个字节 B. 9 个字节 C. 8 个字节 D. 7 个字节

2

字符串的两种常见存储方式

1-字符数组方式

```
char s1[] = "hello";
```

特点:

数组, 空间可写

内容可以修改

存在栈区 (局部)

2-字符指针方式

```
char *s2 = "hello";
```

特点:

指向字符串常量

内容不可修改 ✗

3

字符串的输入与输出

1-输出

```
printf("%s\n", str); \\不自动换行  
puts(str); \\自动换行
```

2-输入

```
scanf("%s", str); // 不能读空格
```

4

字符串与数组的关系

`char str[10];`

str 是数组名

代表首元素地址

不能整体赋值

5

字符串常用库函数



`#include <string.h>`

要包含头文件



1-strlen —— 求长度

```
int len = strlen("hello"); // 5 不包含 '\0'
```

2-strcpy —— 字符串拷贝

```
char dst[20];
strcpy(dst, "hello");
```

3-strcat —— 字符串拼接

```
char s[20] = "hi ";
strcat(s, "there");
```

4-strcmp —— 字符串比较

```
strcmp("abc", "abd"); // < 0
strcmp("abc", "abc"); // = 0
strcmp("abd", "abc"); // > 0
```

字符串遍历

```
char s[] = "hello";
for (int i = 0; s[i] != '\0'; i++) {
    printf("%c ", s[i]);
}
```

字符串反转

```
void reverse(char *s)
{
    int i = 0, j = strlen(s) - 1;
    while (i < j) {
        char t = s[i];
        s[i] = s[j];
        s[j] = t;
        i++; j--;
    }
}
```

字符串与指针的关系

str

首地址

***str**

第一个字符



str + 1

第二个字符地址

***(str+1)**

第二个字符



例题

8、给出定义： char x[]="abcdefg"; char y[]={‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’}；
则正确的叙述为（ C ）

- A. 数组 x 和数组 y 等价
- B. 数组 x 和数组 y 的长度相同
- C. 数组 x 的长度大于数组 y 的长度
- D. 数组 x 的长度小于数组 y 的长度

9、若说明： char s1[]="tree", s2[]{"flower"}；
则以下对数组或数组元素的输出语句中，正确的是（ D ）。

- A. printf("%s%s",s1[3], s2[5]);
- B. printf("%c%c",s1, s2);
- C. puts(s1, s2);
- D. puts(s1); puts(s2);

9.为了判断两个字符串 s1 和 s2 是否相等，应当使用 D 。

- A. if(s1=s2)
- B. if(strcpy(s1,s2))
- C. if(s1==s2)
- D. if (strcmp(s1,s2))

```
5. #include<stdio.h>
#include<string.h>
void fun(char *s[], int n){
    char *t;
    int i, j;
    for(i=0;i<n-1;i++)
        for(j=i+1;j<n;j++)
            if( strlen(s[i]) > strlen(s[j]) ){
                t=s[i]; s[i]=s[j]; s[j]=t;
            }
}
void main(){
    int i;
    char *ss[]={ "hello", "how", "world!", "hi", "Hello world"};
    fun(ss, 5);
    for(i=0;i<5;i++)
        printf ("%s\n", ss[i]);
}
```

hi
how
hello
world!
Hello world!

14. 下列给字符串变量赋值或赋初值，不正确的方式是 C。

- A. char *p; p="string";
- B. char *p="string";
- C. char str[10]; str="string";
- D. char str[]="string";

```
void main(){  
    int i;  
    char str[ ]="2Tia3n @4&j i5n";  
    for(i=0; str[i]!='\0'; i++)  
        if(str[i]>='0' && str[i]<='9')  
            printf("%c\n", str[i]);  
}
```

程序运行后的输出结果是：

2
3
4
5

```
#include<stdio.h>  
#include<string.h>  
void func(char *p, int n){  
    char temp,*p1,*p2;  
    for(p1=p, p2=p+n-1; p1<p2; p1++, p2--){  
        temp=*p1;  
        *p1=*p2;  
        *p2=temp;  
    }  
}  
void main(){  
    char str[]="GoodLuck!";  
    func (str, strlen(str));  
    puts(str);  
}
```

程序运行后的输出结果是：

!kcuLdooG