

字符数组

1. 单选

- 以下对 C 语言字符数组描述错误的是：_____。
 - A) 字符数组可以存放字符串
 - B) 字符数组里的字符串可以进行整体输入输出
 - C) 可以在赋值语句中通过赋值运算符“=”对字符数组整体赋值
 - D) 字符数组的下标从 0 开始
- 以下错误的语句是：_____。
 - A) `char word[] = {'C', 'h', 'i', 'n', 'a'};`
 - B) `char word[] = {"China"};`
 - C) `char word[] = "China";`
 - D) `char word[] = 'China';`
- 已知 `char str1[10], str2[]={"China"};`, 则在程序中能将字符串“China”赋给数组 str1 的正确语句是：_____。
 - A) `str1={"China"};`
 - B) `strcpy(str1, str2);`
 - C) `str1=str2;`
 - D) `strcpy(str2, str1);`
- 以下程序的执行结果是：_____。

```
#include <stdio.h>
int main()
{
    char s[]={'a', 'b', '\0', 'c', '\0'};
    printf("%s\n", s);
    return 0;
}
```

 - A) 'a' 'b'
 - B) ab
 - C) ab c
 - D) 以上都不对
- 已知 `char c1[]={"abcd"}; char c2[]={'a', 'b', 'c', 'd'};`, 则下列叙述正确的是：_____。
 - A) 数组 c1 和数组 c2 等价
 - B) 数组 c1 和数组 c2 的长度相同
 - C) 数组 c1 的长度大于数组 c2 的长度
 - D) 数组 c1 的长度等于数组 c2 的长度
- 以下程序段的运行结果是：_____。

```
char s[] = "\\141\\141abc\\t";
printf("%d\n", strlen(s));
```

 - A) 9
 - B) 12

C) 13

D) 14

- 已知 `char string1[10] = "abcde"`, `string2[10] = "xyz"`, 则以下程序语句的输出结果是: _____。

```
printf("%d", strlen(strcpy(string1, string2)));
```

A) 3

B) 5

C) 8

D) 9

- 下面程序的运行结果是: _____。

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

```
#include <string.h>
```

```
int main()
```

```
{
```

```
    char s[] = "\n123\\";
```

```
    printf("%d,%d\n", strlen(s), sizeof(s));
```

```
    return 0;
```

```
}
```

A) 赋初值的字符串有错

B) 6, 7

C) 5, 6

D) 6, 6

- 下面程序的运行结果是: _____。

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

```
int main()
```

```
{
```

```
    char ch[3][5] = {"AAAA", "BBB", "CC"};
```

```
    printf("%s\n", ch[1]);
```

```
    return 0;
```

```
}
```

A) "AAAA"

B) "BBB"

C) "BBBCC"

D) "CC"

- 以下程序根据用户输入的字符串 `c` 在二维数组 `str` 中顺序查找。请填空:

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

```
#include <string.h>
```

```
int main()
```

```
{
```

```
    char str[][10] = {"Basic", "C++", "FoxPro", "Delphi", "Oracle"};
```

```
    char c[15];
```

```
    int n=5, i=0;
```

```
    scanf("%s", c);
```

```
    printf("c=%s\n", c);
```

```

while(____) i++;
if(i<n)
    printf("str[%d]=%s\n", i, str[i]);
else
    printf("不存在! \n");
return 0;
}

```

- A) i<n && strcmp(str[i], c)!=0
- B) strcmp(str[i], c)!=0
- C) i<n
- D) i<n && str[i]!=c

- 字符数组 s 中的字符已按升序排列，下面的程序将字符串 s1 的每个字符按升序插入到数组 s 中。

```

#include <stdio.h>
int main()
{
    char s[30] = "cehijknqrtuw";
    char s1[] = "fobgvlapdms";
    int i, j, k;
    for(k=0; s1[k] != '\0'; k++)
    {
        j=0;
        while(s1[k] >= s[j] && s[j] != '\0') j++;
        for( (1) ) (2) ;
        s[j] = s1[k];
    }
    puts(s);
    return 0;
}

```

- (1)
 - A) i=strlen(s); i>=j; i--
 - B) i=strlen(s)+k; i>=j; i--
 - C) i=j; i<=strlen(s)+k; i++
 - D) i=j; i<=strlen(s); i++
- (2)
 - A) s[i+1]=s[i]
 - B) s[i]=s[i+1]
 - C) s[i]=s[i-1]
 - D) s[i-1]=s[i]

- 下面程序的功能是将已按升序排列的两个字符串 a 和 b 中的字符按升序归并到字符串 c 中：

```

#include <stdio.h>
int main()
{
    char a[] = "acegikm";
    char b[] = "bdfhjlnpq";

```

```

char c[30];
int i=0, j=0, k=0;
while(a[i]!='\0' && b[j]!='\0')
{
    if(a[i] < b[j])
        (1);
    else
        (2);
}
while(a[i]!='\0') (3);
while(b[j]!='\0') (4);
c[k]='\0';
puts(c);
return 0;
}

```

- (1) A) c[k++]=a[i++]
 B) c[k]=a[i]
 C) a[i]=c[k]
 D) c[k++]=a[i]
- (2) A) c[k]=b[j]
 B) c[k++]=b[j++]
 C) b[j]=c[k]
 D) c[j++]=b[k++]
- (3) A) c[k]=a[k]
 B) a[i]=c[k]
 C) c[k++]=a[i++]
 D) b[j]=c[k]
- (4) A) c[k]=b[j]
 B) c[k++]=b[j]
 C) b[j++]=c[k++]
 D) c[k++]=b[j++]

2. 判断

- 定义语句 `int a[10]={‘A’, ‘B’, ‘C’, ‘D’};`是否正确。
- C 语言中，gets()函数的返回值是用于存放输入字符串的字符数组的首地址。
- 以下程序的结果是：a=4,b=4

```

#include <stdio.h>
#include <string.h>
int main()
{
    int a, b;
    char str[5] = "ABC";
    a=strlen("ABC");
}

```

```

    b=sizeof(str);
    printf("a=%d", a);
    printf("b=%d\n", b);
    return 0;
}

```

- 已知字符数组 str1 的初值为“China”，则语句 `str2=str1`; 执行后字符数组 str2 也存放字符串“China”。
- `char c[] = "Very Good"`; 是一个合法的为字符数组赋值的语句。
- `char s[5] = "abcde"`; 是一个合法的字符数组定义语句。
- 字符处理函数 `strcat(str1, str2)` 的功能是把 str1 接到 str2 的后面。
- 采用 `char str[4][10]`; 语句定义的字符串数组通常用于存放 4 个长度不超过 9 个字符的字符串。
- 在定义语句 `char s[2][10]`; 执行后，可以通过 `s[0]='ABCDE'` 进行赋值操作。
- 在定义语句 `char s[2][10]={"abcde", "12345"}`; 执行后，可以通过 `s[0]`、`s[1]` 来输出两个字符串，如：`printf("%s,%s\n", s[0], s[1]);`。
- `char s[][10]={"abcde", "12345"}`; 是合法的字符串数组定义语句。
- 定义语句 `char d[][4]={"JA", "FE", "MA", "AP", "MA", "JU", "JU", "AU", "SE", "OC", "MO", "DE"}`; 是否正确。
- 在定义语句 `char s[2][10]={"ab", "1234"}`; 执行后，`s[0]`、`s[1]` 的长度都为 10。
- 在定义语句 `char s[2][10]={"ab", "1234"}`; 执行后，`s[0]`、`s[1]` 占用的内存空间分别为 3、5 个字节。

3. 简答

- 下面程序是将某字符串逆序输出，请问，为什么会出现错误的运行结果？

```

#include <stdio.h>
#include <string.h>
int main()
{
    char str[]="abcdef",m;
    unsigned i,j;
    printf("the original string: %s\n", str);
    for(i=0,j=strlen(str);i<strlen(str)/2;i++,j--)
    {
        m=str[i];
        str[i]=str[j];
        str[j]=m;
    }
    printf("the inverse string: %s\n",str);
    return 0;
}

```

- 运行结果：

```
"E:\王化雨\01 教...  — □ ×
the original string: abcdef
the inverse string:
Press any key to continue
```

- 解释下面程序的运行结果：

```
#include <stdio.h>
int main()
{
    char a[2][5]={"China","Jinan"};
    char b[2][6]={"China","Jinan"};
    printf("a[0]=%s,a[1]=%s\n",a[0],a[1]);
    printf("b[0]=%s,b[1]=%s\n",b[0],b[1]);
    return 0;
}
```

```
"E:\王化雨\01 教学\04 C程序设计\01 ...  — □ ×
a[0]=ChinaJinan烫p □ a[1]=Jinan烫p □
b[0]=China, b[1]=Jinan
Press any key to continue_
```

4. 程序运行结果

- 以下程序的运行结果是：_____。

```
#include <stdio.h>
int main()
{
    char ch[3][5] = {"AAAA", "BBBbb", "CC"};
    printf("\n"%s\n", ch[1]);
    return 0;
}
```

- 以下程序的运行结果是：_____。

```
#include <stdio.h>
#include <string.h>
int main()
{
    char s1[40] = "ab", s2[20] = "cdef";
    int i=0;
    strcat(s1, s2);
    while(s1[i] != '\0')
    {
        s2[i] = s1[i];
        i++;
    }
}
```

```

    }
    s2[i] = '\0';
    puts(s2);
    return 0;
}

```

- 以下程序的运行结果是：_____。

```

#include <stdio.h>
#include <string.h>
int main()
{
    char s1[40] = {"some string * "};
    char s2[20] = "test";
    printf("%d ", strlen(s2));
    strcat(s1, s2);
    printf("%s\n", s1);
    return 0;
}

```

- 分析以下程序的执行结果。

```

#include <stdio.h>
int main()
{
    char s1[] = "this book", s2[] = "this hook";
    int i;
    for(i=0; s1[i] != '\0' && s2[i] != '\0'; i++)
        if(s1[i] == s2[i])
            printf("%c", s1[i]);
    printf("\n");
    return 0;
}

```

- 分析以下程序的执行结果。

```

#include <stdio.h>
int main()
{
    char s[] = "china", c;
    int i, j;
    for(i=0; i<5; i++)
        for(j=4; j>=1; j--)
            if(s[j] < s[j-1])
            {
                c=s[j];
                s[j]=s[j-1];
                s[j-1]=c;
            }
    printf("%s\n", s);
}

```

```

    return 0;
}

```

- 分析以下程序的执行结果。

```

#include <stdio.h>
#include <string.h>
int main()
{
    char s[][10]={"while","for","switch","if","break","continue"};
    char temp[10];
    int i, j;
    for(i=0; i<5; i++)
        for(j=5; j>i; j--)
            if(strcmp(s[j], s[j-1])<0)
            {
                strcpy(temp, s[j]);
                strcpy(s[j], s[j-1]);
                strcpy(s[j-1], temp);
            }
    for(i=0; i<6; i++)
        printf("%s\n", s[i]);
    return 0;
}

```

5. 程序完形填空

- 下面程序的作用是输出两个字符串中较短字符串的长度，请填空：

```

#include <stdio.h>
#include <string.h>
int main()
{
    char p1[20], p2[20];
    int i1, i2, num;
    gets(p1); gets(p2);
    i1 = (1);
    i2 = (2);
    num = ( (3) ? (4) : (5) );
    printf("num=%d\n", num);
    return 0;
}

```

- 以下程序对数组中的元素值进行从大到小的排序，请填空。

```

#include <stdio.h>
int main()
{
    int a[] = {2,4,15,3,17,5,8,23,9,7,11,13}, i, j, k;

```



```

    for(k=0; k<11; k++)
    {
        for(i=k+1; i<12; i++)
        {
            if(a[i] (1) a[k])
            {
                j=a[i];
                a[i]= (2) ;
                (3) =j;
            }
        }
    }

    for(k=0; k<12; k++) printf("%4d", a[k]);
    printf("\n");
    return 0;
}

```

- 以下程序采用直接插入排序对数组 R 中的元素值进行从大到小的排序，请填空。

```

#include <stdio.h>
int main()
{
    int R[] = {2,4,15,3,17,5,8,23,9,7,11}, i, j, tmp, n=11;
    for(i=1; i<n; i++)
    {
        if( (1) )
        {
            tmp=R[i]; j=i-1;
            do
            {
                R[j+1]=R[j];
                j--;
            } while(j>=0 && (2) );
            (3) ;
        }
    }
    for(i=0; i<n; i++) printf("%d ", R[i]);
    printf("\n");
    return 0;
}

```

- 下面程序的功能是：将字符数组 a 中下标值为偶数的元素从小到大排列，其他元素不变。请填空。

```

#include <stdio.h>
#include <string.h>
int main()
{
    char a[] = "clanguage", t;
    int i, j, k;

```

```
k=strlen(a);  
for(i=0; i<=k-2; i+=2)  
    for(j=i+2; j<=k; (1))  
        if((2))  
        {  
            t=a[i];  
            a[i]=a[j];  
            a[j]=t;  
        }  
puts(a);  
return 0;  
}
```