

## 主题作业二：数组与

答案：B

解释：（1）多维数组元素按照顺序放置，  
a[0][0]、a[0][1]、a[1][0]、a[1][1]、a[2][0]、a[2][1]  
（2）a[i]是指向&a[i][0]的指针。

### 一、单选题

- 数组定义为 `int a[3][2]={1, 2, 3, 4, 5, 6}`，数组元素 B 的值为 6。  
A、a[3][2]      B、a[2][1]      C、a[1][2]      D、a[2][3]
- 不正确的赋值或赋初值的方式是 C。  
A、`char str[]="string";`  
B、`char str[7]={'s', 't', 'r', 'i', 'n', 'g'};`  
C、`char str[10]; str="string";`  
D、`char str[7]={'s', 't', 'r', 'i', 'n', 'g', '\0'};`
- 设变量定义为 `char s[]="hello\nworld\n"`，则数组 s 中有 B 个元素。  
A. 12      B. 13      C. 14      D. 15
- 下列代码段的输出为 A。  
`char c[]="\\t\\r\\0will\\n";`  
`printf("%d",strlen(c));`  
A. 4      B. 15      C. 16      D. 11
- 下列代码段输出字符串 str，则循环条件 `str[i]!='\0'` 能够被下列哪个选项替换 A。  
`char str[80]="hello, world";`  
`for (i = 0; str[i] != '\0'; i++)`  
`putchar(str[i]);`  
A. `str[i]`      B. `i < 80`      C. `!(str[i] == '\0')`      D. `i <= 80`
- 关于数组声明“`int a[4];`”，下列用法错误的是 D。  
A. `*a`      B. `a[0]`      C. `a`      D. `a++`
- 下列关于数组的定义，哪种用法是错误的 C。  
A. `int a[1][3];`  
B. `int x[2][2]={1,2,3,4};`  
C. `int x[2][]={1,2,4,6};`  
D. `int m[][3]={1,2,3,4,5,6};`
- 以下不正确的定义语句是 D。  
A. `double x[5]={2.0,4.0,6.0,8.0,10.0};`  
B. `char c2[]={'x10','xa','x8'};`  
C. `char c1[]={'1','2','3','4','5'};`  
D. `int y[5]={0,1,3,5,7,9};`
- 若变量已正确定义并且指针 p 已经指向某个变量 x，则 `(*p)++` 相当于 B。  
A、`p++`      B、`x++`      C、`*(p++)`      D、`&x++`
- 若 p1、p2 都是整型指针，p1 已经指向变量 x，要使 p2 也指向 x，A 是正确的。  
A、`p2=p1;`      B、`p2=**p1;`      C、`p2=&p1;`      D、`p2=*p1;`
- 下列程序段的输出是 B。  
`int c[]={1, 3, 5};`  
`int *k=c+1;`  
`printf("%d", *++k);`  
A、3      B、5      C、4      D、6
- 对于如下说明，语法和语义都正确的赋值是 C。  
`int c, *s, a[]={1, 3, 5};`  
A、`c=*s;`      B、`s[0]=a[0];`      C、`s=&a[1];`      D、`c=a;`
- 设变量定义为 `int a[2]={1,3}, p=&a[0]+1;`，则 \*p 的值是 B。  
A. 2      B. 3      C. 4      D. `&a[0]+1`

数组名是不能被修改的指针

字符串以 '\0' 为结束符

字符串的长度为 12，加上结束符 '\0' 一共存放了 13 个元素

数组名是不能被修改的指针

数组的定义只能省略第一个长度值（即变化最慢的维度）

数组初始化的元素个数不能大于其长度

对 p 所指向的变量进行加 1 操作

Int \*k=c+1 使得 k 指向 c+1。  
++k 后 k 指向 c+2。  
所以 \*++k 的值是 5。

A: 指针 s 的地址不可测，无意义  
B: 指针 s 的地址不可测，无意义  
D: 类型不一致，无法赋值

p 定义为指针，指向 a+1

14. 下列程序段的输出结果是\_\_B\_\_。

```
int *p, *q, k = 1, j=10;
p=&j; q = &k; p = q; (*p)++;
printf("%d", *q);
```

A. 1 B. 2 C. 10 D. 11

p=q 之后, p 和 q 都指向变量 k  
(\*p)++相当于 k++,  
所以(\*p)++之后 k 的值为 2

15. 根据声明int a[10], \*p=a;, 下列表达式错误的是\_\_D\_\_。

A. a[9] B. p[5] C. \*p++ D. a++

数组名是指针, 但是不能被修改

16. 下列表达式, 错误的是\_\_A\_\_。

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

数组名是指针, 但是不能被修改

17. 根据声明int a[10], p=&a[1];数组 a 的最后一个元素是\_\_B\_\_。

A. a[10] B. p[8] C. p[9] D. p[10]

18. 下列代码段将会打印出 ( C )

```
char s[ ]="student";
printf("%s%d", s+3, sizeof(s));
```

A. student7 B. dent7 C. dent8 D. student8

运算符 sizeof 作用在数组上时, 结果为数组的元素个数\*每个元素的大小, 即整个数组所占的 byte 数。数组定义时如果没有明确写出长度, 则其长度由初始化的元素个数决定。当用字符串初始化字符数组时, 还要计算结束字符 '\0'。

19. 下列语句定义pf为指向float类型变量f的指针, \_\_B\_\_是正确的。

A. float f, \*pf = f;  
B. float f, \*pf = &f;  
C. float \*pf = &f, f;  
D. float f, \*pf =0.0;

A 错在初始化 pf=f  
C 错在 f 定义在后  
D 错在 pf 初值不能为浮点数

20. 假设scanf语句执行时输入ABCDE<回车>, 能使puts(s)语句正确输出ABCDE字符串的程序段是\_\_D\_\_。

A. char s[5]="ABCDE";puts(s)  
B. char s[5]={'A','B','C','D','E'}; puts(s);  
C. char \*s;scanf("%s",s);puts(s);  
D. char \*s; s="ABCDE";puts(s);

A 错在初始化  
B 错在初始化不是字符串  
C 错在指针 s 的地址不可测, 无意义

## 二. 填空题

21. 根据声明 int a[2][3][4], 该数组元素数量为\_\_24\_\_。

数量为: 2\*3\*4

22. 下列代码段将会打印出\_\_16\_\_。

```
int array [] = {2, 8, 6, 11, 13, 2, 8, 2};
int k;
int total = 0;
for ( k = 0; k < 8; k++ ) {
    if ( array[k] % 2) break;
    total += array[k];
}
printf ( "%d", total );
```

碰到奇数跳出循环

23. 下列代码段的输出为\_\_5, 20\_\_。

```
char st[20]="hello\0world!";
printf("%d,%d\n",strlen(st),sizeof(st));
```

该题数组大小为 20。参考第 18 题的解释。

24. 写出下列程序段的输出结果。

```
char str[]="hello\tworld\n";
printf("%d, %c\n", sizeof(str), *(str+10));
```

13, d  
(参考第 18 题的解释)

25. 下列代码段的输出为\_\_\_\_\_。

```
int c[ ]={10, 30, 5}, *pc;
for(pc=c; pc<c+2; pc++) printf("%d#", *pc);
```

输出为: 10#30#  
注意 c+2 指向元素 5

26. 下列代码段的输出为\_\_\_\_\_。

```
int a = 1;
int b = 2;
int *x, *y;
x = &a;
y = &b;
*x = *x + 1;
x = y;
*x = *x + 1;
printf("%d, %d", a, b);
```

输出为： 2, 3

27. 下列代码段的输出为\_\_\_\_\_。

```
int a[]={1,2,3,4,5,6,7}, *p, s;
for(s=0, p=a+1; p<=a+6; p=p+2)
    s+= *p;
printf("%d", s);
```

a+1 指向元素 2  
a+6 指向元素 7  
循环累加了 2、4、6  
所以输出为： 12

28. 下列代码段

```
int i=0, s=0, a[10];
while((i<10)&&(a[i]>=0))
    s+=a[i++];
```

等价于代码段

```
int s, a[10], *p=a;
for(__s=0; p<a+10; s += *p++ __)
    if(*p<0) break;
```

a+1 指向元素 2  
a+6 指向元素 7  
输出为： 12

29. 函数 f 1 ( ) 等价于函数 f 2 ( )

```
void f1(char s[], char t[])
{
    int i=0;
    while (t[i]!='\0') {s[i]=t[i]; i++;}
    s[i]=0;
}

void f2(char *s, char *t)
{
    while (__*s++ = *t++__);
}
```

30. 下列程序段输出 ef\_\_\_\_\_。

```
char s[]="abc\0def\0ghi", *sp=s;
printf("%s", sp+5);
```

字符串以'\0'为结束符

三、 阅读理解题

31. 若输入 3 1 2 3 <回车> , 则以下程序的输出结果是\_\_\_\_\_。

```
#include "stdio.h"
void main()
{
    int i, j, n, a[10];
    scanf("%d", &n); // n = 3
    for (i=0; i<n; i++)
```

```

scanf("%d",&a[i]); //a[] = {1,2,3}
for (i=0; i<n; i++)
    for (j=0; j<n; j++)
        printf("%d, ", a[(i+j)%n]);
}

```

1, 2, 3, 2, 3, 1, 3, 1, 2,

32. 输入 123+234=357Yes?<回车>后，下列程序的输出结果是\_\_\_\_\_。

```

#include <stdio.h>
void main( )
{
    int k, n1, n3[10];
    char c;
    n1=0;
    for(k=0; k<10; k++)
        n3[k]=0;
    while((c=getchar( ))!='\n'){
        switch(c){
            case'0': case '1': case '2': case '3': case '4':
            case'5': case '6': case '7': case '8': case '9':
                n3[c-'0']++;
                break;
            default:
                n1++;
                break;
        }
    }
    if(n1) printf("%d ", n1);
    for(k=0; k<10; k++)
        if(n3[k]) printf("%d ", n3[k]);
}

```

统计字符 0~9 出现的次数，放入数组 n3 中  
其他字符的出现次数累计在变量 n1 中  
所以 n3[] 中的值为  
{0, 1, 2, 3, 1, 1, 0, 1, 0, 0}  
n1 的值为 6

因此程序输出为：  
6123111

33. 若输入为 7 3 0 3 0 3 1 2 9 7 6 0<ENTER>，则输出为\_\_3# -1#\_\_\_\_\_。

```

#include <stdio.h>
void main( )
{
    int j, k, sub, x;
    int a[5];

    for(j=1; j<=2; j++){
        for(k=0; k<5; k++)
            scanf("%d", &a[k]);
        scanf("%d", &x);
        sub=-1;
        for(k=0; k<5; k++)
            if(a[k]==x) sub=k;
        printf("%d#", sub);
    }
}

```

34. 下列程序的输出为\_\_\_\_\_。

```
#include <stdio.h>
void main()
{
    int i, k, m;
    int a[8]={60,75,63,92,87,64,79,21}, s[10];

    for(k=0; k<10; k++)
        s[k]=0;
    for(i=0; i<8; i++){
        k=a[i]/10;
        s[k]++;
    }
    m=s[0];
    k=1;
    while (k<10){
        if(s[k]!=0) printf("%d#", s[k]);
        if (s[k]>m) m=s[k];
        k++;
    }
    printf("%d", m);
}
```

s[] 中的值为  
{0, 0, 1, 0, 0, 0, 3, 2, 1, 1}  
因此程序输出为:  
1#3#2#1#1#3

35. 若输入 **AabD** <ENTER>, 则下列程序的输出为\_\_\_\_\_。

```
#include <stdio.h>
void main()
{
    char s[81];
    int i=0;
    gets(s);
    while (s[i]!='\0'){
        if(s[i]<='z'&& s[i]>='a')
            s[i]='z'+ 'a'-s[i];
        i++;
    }
    puts(s);
}
```

s[] 中的值为"AabD"  
while 语句把 s 中小写字母(a-z)对称颠倒  
因此程序输出为:  
AzyD

36. 下列程序运行时输入: <回车>, 输出结果为\_\_\_\_\_。

```
#include <stdio.h>
void f(int a[], int n);
main()
{
    int i, n, a[10];

    scanf("%d", &n); //n = 8
    for (i=0; i<n;i++)
```

```

scanf("%d",&a[i]); //a[] = 3 7 2 1 8 3 4 1
f(a+n/4,n/2); //f(a+2,4),把 2、1、8、3 排序为 1、2、3、8
for (i=0; i<n; i++)
    printf("%d#",a[i]);
}

void f(int a[], int n) //从小到大排序
{
    int i, j, temp;

    for( i = 1; i < n; i++)
        for( j = 0; j < n-i; j++ )
            if (a[j] > a[j+1]) { temp = a[j]; a[j] = a[j+1]; a[j+1] = temp; }
}

```

因此程序输出为：  
3#7#1#2#3#8#4#1#

37. 下列程序的输出为\_\_\_\_\_。

```

#include <stdio.h>
int z;
void p(int *x,int y)
{
    ++ *x;
    y ++;
    z = *x+y;
    printf("%d,%d,%d#",*x, y, z);
}
void main()
{
    int x=1, y=2, z=3;
    p(&y, x);
    printf("%d,%d,%d#",x, y, z);
}

```

程序输出为：  
3, 2, 5#1, 3, 3#

38. 下列代码段的输出为\_42\_\_\_\_\_。

```

#include <stdio.h>
int f(int a[], int n, int m)
{
    int i, j, sum, max=0;
    for(j=0; j<m; j++) max+=a[j];
    for (i=1; i<n; i++) {
        sum=0;
        for (j=0; j<m; j++)
            sum+= *(a+i*m+j);
        if (sum>max) max=sum;
    }
    return max;
}
void main()
{

```

f 函数把 a 看作是[n][m]的数组  
计算并返回最大的行和

```

        static int a[]={1,2,3,4,5,6,7,8,9,10,11,12};
        printf("%d", f(a, 3, sizeof(a)/sizeof(int)/3)); //f(a,3,4)
    }

```

39. 下列程序运行时输入: **C Java C++ VB Fortran Basic** <回车>, 输出结果为 Fortran。

```

#include <stdio.h>
void find(char *str)
{
    char *p=str;
    int k1=0,k2=0;

    while(1) {
        if(*p!=' ' && *p!='\t' && *p!='\0') k2++; //单词以空格，制表符，'\0'为分隔符
        else {
            if (k2>k1) { k1=k2; str=p-k2;} //找到一个更长的单词
            k2=0;
            if (*p=='\0') break;
        }
        p++;
    }
    for(k2=0;k2<k1;k2++) printf("%c",*str++); //输出该最长的单词
    return;
}

main()
{
    char a[80];
    gets(a);
    find(a);
}

```

40. 下列代码段的输出为 0, 4。

```

#include <stdio.h>
void p(int *x,int y)
{
    ++ *x;
    y=y+2;
}
void main()
{
    int x=0, y=3;
    p(&y, y);
    printf("%d, %d", x, y);
}

```