

期中测试回顾

Review

Daomiao Wang (王道淼)
Fall Semester, 2022

平均分： 36.14/65， 13.04/35， 48.97/100

中位数： 36， 15， 51

及格人数	15
90+	1
80-89	2
70-79	5
60-69	7
50-59	20
40-49	15

1.1 设定义数组s的代码为char s[]="x\148dx\t\\\n"; 则s的元素个数是__9__

x\148dx\t\\\n→ x, \14, 8, d, x, \t, \\, \n, \0

```
#include <stdio.h>
int main(void)
{
    char s[] = "x\148dx\t\\\n";
    for (int i = 0; s[i] != '\0'; i++)
    {
        printf("%d element is %c\n", i, s[i]);
    }
    return 0;
}
```

Microsoft Visual Studio Debug Console

```
0 element is x
1 element is 
2 element is 8
3 element is d
4 element is x
5 element is 
6 element is \
7 element is
```

转义字符常量用'\'字符'或'字符列'来标记。例如，换行符用'\n'标记，水平制表符用'\t'标记，回车符用'\r'标记。常用的转义字符及其含义如表 2-1 所示。

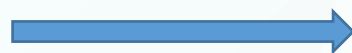
表 2-1 转义字符及其含义

转 义 字 符	含 义	转 义 字 符	含 义
\a	响铃	\\	反斜杠符\
\b	退格（【 Backspace 】键）	\'	单引号符'
\n	换行符，光标位置移到下一行首	\"	双引号符"
\r	回车符，光标位置移到当前行首	\0	字符串结束符
\t	水平制表符（【 Tab 】键）	\ddd	ddd 为 1~3 个八进制数字。如'\12'也能表示换行符'\n'
\v	竖向跳格符	\xhh	hh 为 1~2 个十六进制数字。如'\x41'也能表示大写字母'A'
\f	走纸换页		



1.2 执行语句 “for(j = k = 0; j - 5; k += j++);”后k的值是 10

```
int main(void)
{
    int j, k;
    for (j = k = 0; j - 5; k += j++)
        printf("In loop %d, k is %d\n", j, k);
    printf("\nk is %d\n", k);
    return 0;
}
```



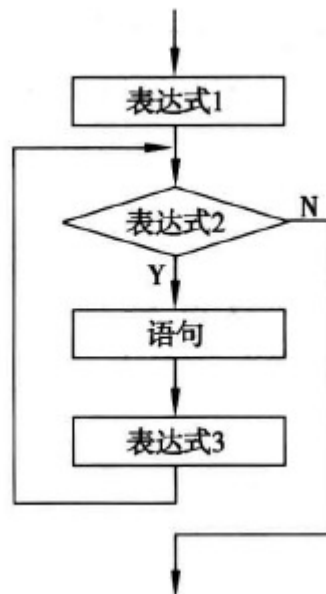
```
int main(void)
{
    int j, k;
    for (j = k = 0; j - 5; j++)
    {
        k += j;
        printf("In loop %d, k is %d\n", j, k);
    }
    return 0;
}
```

```
/*1-2*/
int main(void)
{
    int j, k;
    for (j = k = 0; j - 5; k += j++)
        printf("In loop %d, k is %d\n", j, k);
    printf("\nk is %d\n", k);

    return 0;
}
```

Microsoft Visual Studio Debug Console

```
In loop 0, k is 0
In loop 1, k is 0
In loop 2, k is 1
In loop 3, k is 3
In loop 4, k is 6
k is 10
```



优先级	运算符	类型	结合顺序
1	() [] -> .	伪运算符	→ 自左向右
2	++ -- + - ! ~ * & (type) sizeof	单目运算符	← 自右往左⚠
3	* / %	双目运算符	算术运算 → 自左往右
4	+ -	双目运算符	算术运算 → 自左往右
5	<< >>	双目运算符	移位运算 → 自左往右
6	<= >= < >	双目运算符	比较运算 → 自左往右
7	== !=	双目运算符	相等测试 → 自左往右
8	&	双目位运算符	按位与 → 自左往右
9	^	双目位运算符	按位异或 → 自左往右
10		双目位运算符	按位或 → 自左往右
11	&&	双目逻辑运算符	逻辑与 → 自左往右
12		双目逻辑运算符	逻辑或 → 自左往右
13	?:	三目运算符	条件运算符 → 自左往右
14	= += -= *= /= %= <<= >>= &= ^= =	双目运算符	赋值运算符 ← 自右往左⚠
15	,	逗号运算符	→ 自左往右

1.3 设 `int x = 3;` 表达式 `x += x * x + 1` 的值是 24

1. `(x + 1) := a` 单目运算符

2. `x *= a` -> `x = x * a = 3 * 4 = 12` `:= b` 双目运算符(赋值运算)

3. `x += b` -> `x = x + b = 12 + 12 = 24` 双目运算符(赋值运算)

```
/*1-3*/
int main(void)
{
    int x=3;
    printf("x is %d\n", x+=x*x+1);
    return 0;
}
```

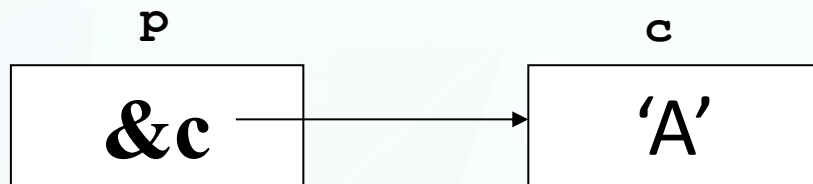
Microsoft Visual Studio Debug Console

x is 24

D:\05_Code_Microsoft_VisualStudio\mid-test-

优先级	运算符	类型		结合顺序
1	() [] -> ,	伪运算符		→ 自左向右
2	++ -- + - ! ~ * & (type) sizeof	单目运算符		← 自右往左⚠
3	* / %	双目运算符	算术运算	→ 自左向右
4	+ -	双目运算符	算术运算	→ 自左向右
5	<< >>	双目运算符	移位运算	→ 自左向右
6	<= >= < >	双目运算符	比较运算	→ 自左向右
7	== !=	双目运算符	相等测试	→ 自左向右
8	&	双目位运算符	按位与	→ 自左向右
9	^	双目位运算符	按位异或	→ 自左向右
10		双目位运算符	按位或	→ 自左向右
11	&&	双目逻辑运算符	逻辑与	→ 自左向右
12		双目逻辑运算符	逻辑或	→ 自左向右
13	?:	三目运算符	条件运算符	→ 自左向右
14	= += -= *= /= %= <<= >>= &= ^= =	双目运算符	赋值运算符	← 自右往左⚠
15	,	逗号运算符		→ 自左向右

1.4 设char *p, c = 'A'; 根据下列图示, 写出建立指针变量p和变量c之间的关系
的赋值语句__p=&c__



1. 指向变量的指针, 使用指针描述数据之间的间接引用关系;
2. 程序用变量名引用变量的值或者变量的地址(程序都是按照变量地址取变量值);
3. 变量地址也是一种数据, 可以存储和运算, 取变量地址的运算符&, 存储程序对象地址的变量成为**指针变量**, 如果指针变量B存储着变量A的地址, 则称指针变量B指向指针变量A;
4. 指针变量的一般形式为 **类型*标识符**;

```
int main(void)
{
    char* p, c = 'A';
    p = &c;
    printf("The ADDRESS of p is %p\n", p);
    printf("The VALUE of p is %c\n", *p);
    return 0;
}
```

Microsoft Visual Studio Debug Console

```
The ADDRESS of p is 000000DC7F31FB14
The VALUE of p is A
```

issues found

D:\05_Code_Microsoft_VisualStudio\mid-test-revi
To automatically close the console when debuggi



1.5 描述 “x/y大于5，并要求y不等于0”的C表达式是 y&&double(x)/y>5 ;
(y!=0)&&double(x)/y>5
y&&x>y*5

逻辑运算中，逻辑非“!”的优先级最高，其次是逻辑与“&&”，逻辑或“||”的优先级最低。另外，逻辑与“&&”和逻辑或“||”的优先级低于关系运算符的优先级；逻辑非“!”的优先级高于算术运算符的优先级。

表 2-8 是逻辑运算真值表。当操作数 a 和 b 分别取不同组合值情况下，各种逻辑运算后的结果。

表 2-8 逻辑运算真值表

a	b	!a	!b	a&&b	a b
非 0	非 0	0	0	1	1
非 0	0	0	1	0	1
0	非 0	1	0	0	1
0	0	1	1	0	0

1.6 设有定义： `int x=5,y=6;int z=(++x>=y++)? x : y-1;` 试给出z的值为 6

条件运算： 表达式1? 表达式2: 表达式3 →

表达式1真，则计算表达式2； 否则计算表达式3

`int x = 5, y = 6;`

`int z = (++x >= y++) ? x : y - 1;`

`(++x >= y++) ? x : y - 1;`

`++x >= y++` -> `5 >= 7` **X** -> `x = 6, y = 7`

`z = y - 1 = 6`

```
/*1-6*/
int main(void)
{
    int x = 5, y = 6;
    int z = (++x >= y++) ? x : y - 1;
    printf("The value of X is %d\n", x);
    printf("The value of Y is %d\n", y);
    printf("The value of Z is %d\n", z);
    return 0;
}
```

Microsoft Visual Studio Debug

The value of X is 6
The value of Y is 7
The value of Z is 6

1.7假定所有变量均已正确定义，下列程序段运行后 x 的值是_4_

```
a = b = c = 0 ; x = 35 ;
```

```
if(!a)x-- ; else if(b); if(c)x = 3; else x = 4;
```

```
int main(void)
{
    int a, b, c, x;
    a = b = c = 0;
    x = 35;
    if (!a) x--;
    else if (b);

    if (c) x = 3;
    else x = 4;

    printf("The value of x is %d", x);
    return 0;
}
```

```
/*1-7*/
int main(void)
{
    int a, b, c, x;
    a = b = c = 0;
    x = 35;
    if (!a) x--;
    else if (b);
    if (c) x = 3;
    else x = 4;
    printf("The value of x is %d", x);
    return 0;
}
```

Microsoft Visual Studio Debug Console

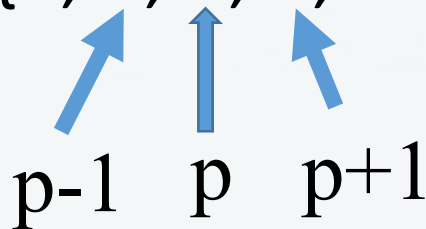
The value of x is 4
D:\05_Code_Microsoft_VisualStudi

1.8变量定义: `int a[]={2, 4, 6, 8, 10}, *p=a+2;` 计算表达式`(p+1)[1]**(p-1)`的值 40

1. 指向数组元素的指针，一维数组名表达式时数组首元素指针, $p=a \sim p[0]=a[0]$
2. 指向数组元素的指针可以做有限的运算:
 - 1). 可以与整数进行加减运算; $*(p+i) = a[i]$
 - 2). 两个指向同一数组的指针可以做大小比较;
 - 3). 两个指向同一数组的指针可以做加减运算。
3. 与用数组名和下标引用数组运算的标记法一致, $(p+i) \sim p[i]$; 若 $p=a[10], p[-1]=p[9]$

`int a[]={2, 4, 6, 8, 10}`

$p-1$ p $p+1$



$(p+1)[1]**(p-1) := (p+1)[1]*(*(p-1))=10*4=40$

```
/*1-8*/
int main(void)
{
    int a[6] = {2, 4, 6, 8, 10}, *p;
    p = a + 2;
    printf("The value of is %d", (p+1)[1]**(p-1));
    return 0;
}
```

Microsoft Visual Studio Debug Console

The value of is 40
D:\05_Code_Microsoft_VisualStudio\mid-test-review\x64\I



2.1对数组中的元素进行排序， 负数在前，非负数在后

```
int a[n]
int i = 0, j = n, temp;
while (i < j)
{
    while (a[i] < 0) i++;
    while (a[j] >= 0) j--;
    if (i < j)
    {
        temp = a[i];
        a[i] = a[j];
        a[j] = temp;
    }
}
```

```
/*2-1*/
int main(void)
{
    int a[10] = { 1, -1, 2, -2, 3, -3, 4, -4, 5, -5 };
    printf("Original array is");
    for (int k = 0; k < 10; k++)
        printf("%3d", a[k]);
    int i = 0, j = 9, temp;
    while (i < j)
    {
        while (a[i] < 0) i++;
        while (a[j] >= 0) j--;
        if (i < j)
        {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
    printf("\nProcessed array is");
    for (int k = 0; k < 10; k++)
        printf("%3d", a[k]);
}
```

Microsoft Visual Studio Debug Console

```
Original array is 1 -1 2 -2 3 -3 4 -4 5 -5
Processed array is -5 -1 -4 -2 -3 3 4 2 5 1
D:\05_Code_Microsoft_VisualStudio\mid-test-review\x64\Deb
To automatically close the console when debugging stops,
le when debugging stops.
Press any key to close this window . . .
```



2.2 程序阅读题

```
int a[6] = { 7,6,8,3,5,4 };
int i, k, temp;
for(i=1;i<5;i++)
    for(k=1;k<4;k++)
        if (a[k] > a[k + 1])
        {
            temp = a[k];
            a[k] = a[k + 1];
            a[k + 1] = temp;
        }
```

} a[1]→a[4]的从小到大的排序

```
for (i = 0; i <= 5; i++)
    printf("%d%c", a[i], i < 5 ? ',' : '\n');
```

格式化输出

```
/*2-2*/
int main(void)
{
    int a[6] = { 7,6,8,3,5,4 };
    int i, k, temp;
    for(i=1;i<5;i++)
        for(k=1;k<4;k++)
            if (a[k] > a[k + 1])
            {
                temp = a[k]; a[k] = a[k + 1]; a[k + 1] = temp;
            }
    for (i = 0; i <= 5; i++)
        printf("%d%c", a[i], i < 5 ? ',' : '\n');
}
```

C:\ Microsoft Visual Studio Debug Console

7, 3, 5, 6, 8, 4

D:\05_Code_Microsoft_VisualStudio\mid-test-review\x64\Debug\mid-test-rev



2.3 程序阅读题：从pos位置起，复制len个字符到字符数组str中，其输出为CDE

```
char str[10], * s1 = str, * s2 = "ABCDEFGG";  
int i = 0, pos = 2, len = 3;  
if (pos < strlen(s2))  
for (s2 += pos; *s2 && i < len; i++)  
    *s1++ = *s2++;  
*s1 = '\\0';  
printf("%s", str);
```

```
/*2-3*/  
int main(void)  
{  
    char str[10], * s1 = str, * s2 = "ABCDEFGG";  
    int i = 0, pos = 2, len = 3;  
    if (pos < strlen(s2))  
        for (s2 += pos; *s2 && i < len; i++)  
            *s1++ = *s2++;  
    *s1 = '\\0';  
    printf("%s", str);  
}
```

Microsoft Visual Studio Debug Console

CDE

D:\05_Code_Microsoft_VisualStudio\mid-test-review\x64\Debug\mid-test-review.exe (C)
To automatically close the console when debugging stops, enable Tools->Options->D
le when debugging stops.
Press any key to close this window . . .



3.1 改错题

以下程序代码是希望实现注释的目的，若不能正确实现，请改正之。

```
/* 1 */ char s[20], *chPt; /* 定义字符数组s和字符指针变量chPt */
```

```
/* 2 */ s = "ABCDE"; /* 给字符数组s赋值字符串"ABCDE" */
```

```
/*2*/ strcpy(s, "ABCDE");
```

```
/* 3 */ *chPt = "1234"; /* 让字符指针变量chPt指向字符串"1234"的首字符*/
```

```
/*3*/ chPt = "1234"
```


3.2 改错题

以下程序代码为把一个字符串中的大写字母变为小写字母，并输出结果。

```
/* 4 */ char s[120], *str=s;
```

```
/* 5 */ scanf("%s", str);
```

```
/* 6 */ while(str){
```

```
/* 6 */ while(*str){
```

```
/* 7 */ if(*str>='A' || *str<='Z')
```

```
/* 7 */ if(*str>='A' && *str<='Z')
```

```
/* 8 */ *str += 'a';
```

```
/* 8 */ *str += 'a' - 'A';
```

```
/* 9 */ str++; }
```

```
/* 10 */ printf("%s", str);
```

```
/* 10 */ printf("%s", s);
```

3.3 改错题

以下程序段是求数组x的前k个整数的和。

```
/* 11*/ int x[]={1,2,3,4,5,6,7,8},int k, i, s=0;
```

```
/* 12*/ scanf("%d",&k);
```

```
/* 13*/ for(i=0;i<=k;i++)
```

```
/* 13 */ for(i=0;i<k;i++)
```

```
/* 14*/ s+=*x;
```

```
/* 14 */ s+=x[i] 或者 s+=*(x+i)
```

4.1 完全程序题

改进的“冒泡法”排序程序段。

```
for( i=0; i<n; i++) scanf("%d", &a[i]);  
m = n-1; /* 为一轮冒泡排序设定上界的初值 */  
while(____m____){  
    for(j=0, i=0; i<m; i++)  
if( a[i]>a[i+1] ){ t=a[i]; a[i]=a[i+1]; a[i+1]=t; j=i; }  
    m = j____;  
}
```

原始序列 :	25	14	54	63	44	3
第一次比较 :	14	25	54	63	44	3
第二次比较 :	14	25	54	63	44	3
第三次比较 :	14	25	54	63	44	3
第四次比较 :	14	25	54	44	63	3
第二次比较 :	14	25	54	44	3	63

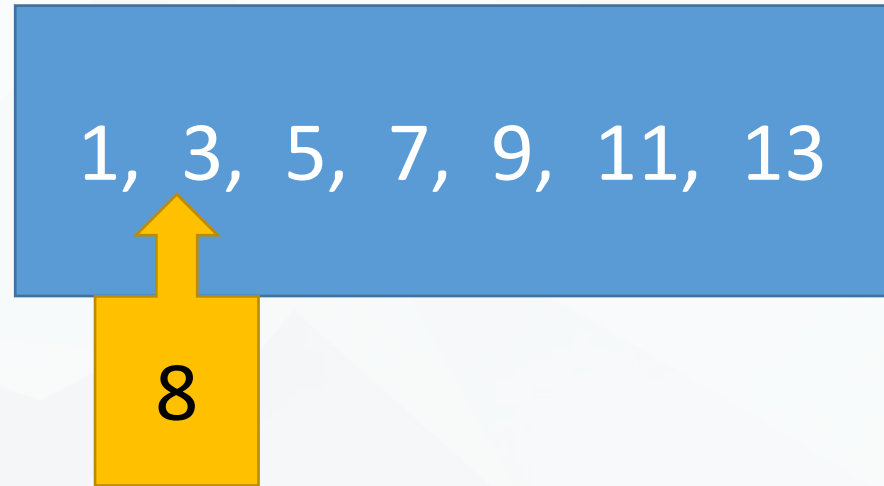
<https://blog.csdn.net/d013028621>



4.2 完全程序题

以下程序段用来在w数组中插入x,数组w中的数据为从键盘输入按从小到大的顺序排列的整数,插入后数组w中的整数仍有序。

```
int w[100],n,x,i,p;  
printf("输入数组元素个数(<100):");scanf("%d",&n);  
printf("按从小到大的顺序输入数组元素值: ");  
for(i=0;i<n;i++) scanf("%d",&w[i]);  
printf("输入插入值: ");scanf("%d",&x);  
p=0; w[n]=x;  
while(x>w[p]) p++;  
for(i=n; i>p; i--)  
    w[i]= w[i-1];  
w[i]=x或w[p]=x ;  
n++;
```



1. 基本数据类型、标准输入输出、运算式、表达式;
 2. 2种选择结构、3种循环结构
 3. 数组与字符串
 4. 函数、递归函数、作用域、宏定义
-

1. 指针 (C和指针)
 2. 结构体与链表 (数据结构)
-

1. 文件 (操作系统接口)





Thanks for your attention!