

计算机程序设计

复习用测试题

本文档含 4 套测试题及详细的答案，均为往年期末考试真题，供科大学生复习和自测使用，勿外传。

自 2025 年起，本课程理论课缩减为 40 学时，期末笔试的考核范围、考试题型进行了调整：

1. 侧重基本概念的理解、阅读程序的能力、模块设计能力的考核，降低纯语法知识的考核比例。
2. 考试题型包括单项选择、多项(不定项)选择、读程序写结果、填空与程序填空、画流程图，不再考手写程序题。
每年的题型可能会有一定的变化。
3. 链表不在笔试考试范围内，以下测试题有的包含了链表的内容，不感兴趣的同学可以忽略。

综合测试题（1）

一、单选题

1. 与数学公式 $3x^n/(2x-1)$ 对应的 C 语言表达式是（ ）。
A. $3*x^n/(2 * x-1)$ B. $3*x** n/(2*x-1)$
C. $3*pow(x,n)/(2*x-1)$ D. $3*pow(n,x)*(2*x-1)$
2. 设 a 为整型变量，不能正确表达数学关系 $10 < a < 15$ 的 C 语言表达式为（ ）。
A. $10 < a < 15$ B. $a == 11 || a == 12 || a == 13 || a == 14$
C. $a > 10 \&\& a < 15$ D. $!(a <= 10) \&\& !(a >= 15)$
3. 下列关于 if 语句结构说法错误的是（ ）。
if(表达式)语句 1; else 语句 2;
A. “表达式”可以是任何表达式
B. 当语句 2 为空时，“else 语句 2;”可以省略
C. 语句 1 或语句 2 可以是复合语句，不可以是空语句
D. else 总是与同一语法层次中离它最近的尚未配对的 if 配对
4. 若已定义 x 和 y 为 double 型, $x=1$, 则表达式 $y = x + 3/2$ 的值是（ ）。
A. 1 B. 2.0 C. 2 D. 2.5
5. 若 `int a[]={1,2,3,4,5,6,7,8,9,10}`, `*p=a`; 则值为 3 的表达式是（ ）。
A. `p+=2, *p++` B. `p+=2, *++p`
C. `p+=3, *p++` D. `p+=2, ++*p`
6. 设 `int(*ptr)[10]`; 其中的 ptr 是（ ）。
A. 10 个指向整型变量的指针
B. 指向 10 个整型变量的函数指针
C. 一个指向具有 10 个元素的一维数组的指针
D. 具有 10 个指针元素的整型数组

7. 设有定义:double x;则以下正确的输入语句是 ()。
- A. scanf("%f", x) B. scanf("%lf", &x)
C. scanf("%lf", x) D. scanf("%lf", %x)
8. 字符串常数 "xiaoli" 在内存占用的字节数是 ()。
- A. 6 B. 7 C. 8 D. 32
9. C 程序由函数组成, 关于 C 语言函数的说法错误的是 ()。
- A. 除整型函数外, 其他类型函数必须在定义时给以类型说明
B. 函数原型说明语句必须给出函数类型的说明
C. 函数不仅可以递归调用, 也可以递归定义
D. 函数类型决定返回值的类型
10. 以下调用 scanf 函数给变量 a 输入数值的方法是错误的, 其原因是 ()。
- ```
int main() {
 int a, *p, *q, b; p = &a;
 printf("input a:");
 scanf("%d", *p);

}
```
- A. \*p 表示的是指针变量 p 的地址  
B. \*p 表示的是变量 a 的值, 而不是变量 a 的地址  
C. \*p 表示的是指针变量 p 的值  
D. \*p 只能用来说明 p 是一个指针变量

## 二、填空题

1. 变量存储类别, 从作用域角度分有\_\_\_\_\_变量和\_\_\_\_\_变量。
2. 设整型变量 x, y, z 均为 5; 执行 x-=y-z 后, x=\_\_\_\_\_。
3. 设 m, n, a, b, c, d 均为 0, 执行 ( m = a == b ) || ( n = c == d ) 后, m, n 的值分别为: \_\_\_\_\_, \_\_\_\_\_。
4. 若有定义: int a=1, b=2;  
执行 printf("%d\n", (a = ++ b, a + 5 , a/5)); 的结果为\_\_\_\_\_。
5. 设 ch 是 char 型变量, 其值为 'A', 则执行语句  
ch = (ch >= 'A' && ch <= 'Z') ? (ch + 32) : ch; 后, ch 的值为\_\_\_\_\_。
6. 若有定义: int x=3, y=5;  
执行 printf("%d\n", (x >= y >= 2)) ? 1 : 0); 的结果为\_\_\_\_\_。
7. 若定义 i, j 为 int 型, 则一下程序段中内循环体的执行次数为\_\_\_\_\_。
- ```
for (i = 5; i; i--) {  
    for (j = 0; j < 4; j++) { ... }  
}
```
8. 设有定义语句
Struct {
 int x;
 int y;
} d[2] = {{1, 3}, {2, 7}};
则 printf("%d\n", d[0].y / d[0].x * d[1].x) 的输出是_____。

三、阅读程序后写出运行结果

1.

```
#include <stdio.h>
int main()
{
    int a, b;

    for (a = b = 1; a <= 100; a = a + 1) {
        if (b >= 10) {
            printf("%d\n", b);
            break;
        }
        if (b % 3 == 1) {
            printf("%d\n", b = b + 3);
            continue;
        }
    }
    printf("%d\n", a);
    return 0;
}
```

2.

```
#include <stdio.h>
int main()
{
    int i = 0, j, n = 7788, base = 8, num[20];
    do {
        i++;
        num[i] = n % base;
        n = n / base;
    } while (n != 0);
    for (j = i; j >= 1; j--) {
        printf("%d", num[j]);
    }
    printf("\n");
    return 0;
}
```

3.

```
#include <stdio.h>
int n = 2;
int fun(int n) {
    int a = 1;
    static int m = 5;
    m = n + m + a;
    printf("%4d%4d\n", n, a);
    return (m);
}
```

```

int main() {
    int a = 1;
    printf("%4d\n", fun(n + fun(a)));
    return 0;
}

```

4.

```

#include <stdio.h>
int main() {
    int k = 1;
    char c = 'A';
    do {
        switch (++c) {
            case 'A': k++; printf("%4d", k); break;
            case 'B': k--; printf("%4d", k);
            case 'C': k += 2; printf("%4d", k); break;
            case 'D': k % 2; printf("%4d", k); break;
            case 'E': k = k * 2; printf("%4d", k); break;
            default: k = k / 3; printf("%4d", k);
        }
    } while (c < 'E');
    printf("\n");
    return 0;
}

```

5.

```

#include <stdio.h>
void write_it(char *s) {
    printf("%s\n", s);
    if (*s != 'a') {
        write_it(s + 1);
    }
    putchar(*s);
}
int main()
{
    char str[40] = "image";
    write_it(str);
    printf("\n");
    return 0;
}

```

6.

```

#include <stdio.h>
#define SWAP(x, y) temp = x; x = y; y = temp
void swap(int *p, int *q)
{
    int temp;
    temp = *p; *p = *q; *q = temp;
}

```

```
int main() {
    int a = 1, b = 3, *p = &a, *q = &b, *temp;
    printf("%d,%d\n", *p, *q);
    SWAP(p, q);
    printf("%d,%d\n", *p, *q);
    swap(p, q);
    printf("%d,%d\n", *p, *q);
    return 0;
}
7.
```

```
#include <stdio.h>
int main(){
    int a[2][3] = {{1, 2, 3}, {4, 5, 6}};
    int b[3][2] = {{3, 2}, {2, 0}, {1, 4}};
    int i, j, k, sum = 0, c[2][3];
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 2; j++) {
            for (c[i][j] = k = 0; k < 3; k++) {
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 2; j++) {
            printf("%d\t", c[i][j]);
            if (i == j || i + j == 1) {
                sum += c[i][j];
            }
        }
        printf("\n");
    }
    printf("%d\n", sum);
    return 0;
}
```

四、程序填空题

1. 下面程序的功能是通过调用函数 f 计算代数多项式当 $a=1.7$ 时的值。

代数多项式为： $1.1+2.2*a+3.3*a*a+4.4*a*a*a+5.5*a*a*a*a$ ，根据程序功能填空。

```
#include <stdio.h>
double f(double x, (1)_____) {
    double y = a[0], t = 1;
    int i;
    for (i = 0; i < n - 1; i++) {
        t = (2)_____;
        y = y + a[i + 1] * t;
    }
    return ((3)_____);
}
```

```
int main() {
    double b[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
    printf("%lf\n", f(1.7, b, 5));
    return 0;
}
```

2. 函数 YangHui 的功能把杨辉三角形的数据赋给二维数组的下半三角，形式如框内所示，其构成规律是：

- (1) 第 0 列元素和主对角线均为 1。
- (2) 其余元素为其左上方和正上方元素之和。
- (3) 数据的个数每行递增 1。

请将程序补充完整。

```
#define N 6
void YangHui(int x[N][N]) {
    int i, j;
    x[0][0] = 1;
    for (i = 0; i < N; i++) {
        x[i][0] = (4) _____ = 1;
        for (j = 1; j < i; j++) {
            x[i][j] = (5) _____;
        }
    }
}
```

1				
1	1			
1	2	1		
1	3	3	1	
1	4	6	4	1
...

五、画流程图题

有移位编码算法函数，对一个无符号整数的各位数字按如下规则进行编码：0 编码为 1，1 编码为 2，…，8 编码为 9，9 编码为 0，如果最高位为 9 则保持其不变，编码后的数据作为返回值。请用流程图描述以上算法。

综合测试题（2）

一、单选题

1. 计算机可以进行自动处理的基础是（ ）。
A. 存储程序 B. 快速运算 C. 计算精度高 D. 能进行逻辑判断
2. 计算机的通用性使其可以求解不同的算术和逻辑问题，这主要取决于其（ ）。
A. 高速运算 B. 指令系统 C. 可编程性 D. 存储功能
3. 在计算机内部用于表示数据和指令的是（ ）编码。
A. 十进制码 B. 二进制码 C. ASCII 码 D. 汉字编码
4. 计算机进行数值计算时的高精度主要取决于（ ）。
A. 计算速度 B. 内存容量 C. 外存容量 D. 基本字长
5. 已知字母 A 的 ASCII 码为十进制数 65，且 c2 为字符型，则执行以下语句 `c2 = 'A' + '6' - '3';` 后，c2 中的值为（ ）。
A. D B. 68 C. 不确定 D. 'C'
6. 设已有定义：`char a[5], *p = a;` 则下面的选项中正确的赋值语句是（ ）。
A. `p = "abcd";` B. `a = "abcd";` C. `*p = "abcd";` D. `*a = "abcd";`
7. 以下错误的描述是（ ）。
A. `break` 语句不能用于循环语句和 `switch` 语句外的任何其他语句
B. 在 `switch` 语句中使用 `break` 和 `continue` 语句的作用相同
C. 在循环语句中使用 `continue` 语句是为了结束本次循环
D. 在循环语句中使用 `break` 语句是为了使流程跳出循环体，提前结束循环。
8. 设有定义：`double x;` 则以下正确的输入语句是（ ）。
A. `scanf("%f", x);` B. `scanf("%f", &x);`
C. `scanf("%lf", &x);` D. `scanf("%lf", x);`
9. 以下对于运算符优先顺序的描述中正确的是（ ）。
A. 关系运算符<算术运算符<赋值运算符<逻辑与运算符
B. 逻辑与运算符<关系运算符<算术运算符<赋值运算符
C. 赋值运算符<逻辑与运算符<关系运算符<算术运算符
D. 算术运算符<关系运算符<赋值运算符<逻辑与运算符
10. C 程序由函数组成，关于 C 语言函数的说法错误的是（ ）。
A. 除整型函数外，其他类型函数必须在定义时给以类型说明。
B. 函数原型说明语句必须给出函数类型的说明。
C. 函数类型决定返回值的类型。
D. 函数不仅可以递归定义，而且除 `main` 函数外都可以递归调用。

二、填空题

1. C 的数据类型使编译器能够确定在内存中如何存储一个特定的值，以及对该数值可以_____。
2. 变量名必须以字母或是下划线开始，并由字母、数字和下划线组成，用户标识符不能与_____同名。
3. 设已有 double T, t; 说明，试将等式 $T = \frac{4t^2}{t+2} - 20$ 写成一条 C 语句_____。
4. 表达式 `1 && 30 % 10 >= 0 && 30 % 10 <= 3` 的值为：_____。
5. 设已有 `int x = 3, y = 4`; 说明，表达式 `(x = ++y, x + 5, x / 5)` 的值为：_____。
6. 设已有 `int j = 5, k = 2, n, m, p`; 说明，执行以下三条语句后 n, m 和 p 的值是什么?
`n = j - ++k; m = j-- + k--; p = k + j;`
解: n 的值是_____, m 的值是_____, p 的值是_____。
7. 设已有 `int x = 3, y = 5, z = 2`; 说明，执行以下三条语句后 x, y 和 z 的值是什么?
`x *= y + z; y /= 2 * z + 1; z += x;`
解: x 的值是_____, y 的值是_____, z 的值是_____。
8. 如果 n 的值是 4, 并且 m 的值是 5, 表达式 `++(n * m)` 的值是 21 吗? 解释你的答案: _____。

三、阅读程序后写出运行结果

1.

```
#include<stdio.h>
int main() {
    int i = 1, x = 1;
    while (++i <= 20) {
        if (x > 6) break;
        else if (x % 2 == 1) {
            x += 5;
            printf("i=%d,x=%d\n", i, x);
            continue;
        }
        x -= i;
        printf("i=%d,x=%d\n", i, x);
    }
}
```
2.

```
#include<stdio.h>
#define PRODUCT(a, b) a * b
void fun(int n) {
    static int x = 1;
    printf("x=%d\n", x + n);
    x += PRODUCT(x + n, x - n);
}
int main() {
    int i, x = 1;
    for (i = 1; i <= 3; i++, x++) {
        fun(x + i);
    }
    return 0;
}
```

3. #include<stdio.h>

```
void udf_sort(int array[ ], int n) {
    int i, j, k, temp;
    for (i = 0; i < n - 1; i++) {
        k = i;
        for (j = i + 1; j < n; j++) {
            if (array[j] < array[k]) {
                k = j;
            }
        }
        if (k != i) {
            temp = array[k];
            array[k] = array[i];
            array[i] = temp;
        }
        for (j = 0; j < n; j++) {
            printf("%d%c", array[j], (j + 1) % 5 ? ' ' : '\n');
        }
    }
}

int main()
{
    int i;
    int a[5] = {2, 5, 3, 1, 4};
    udf_sort(a, 5);
    for (i = 0; i < 5; i++) {
        printf("%d%c", a[i], (i + 1) % 5 ? ' ' : '\n');
    }
    Return 0;
}
```

4. #include<stdio.h>

```
int multiply(int m, int n) {
    int ans;
    if (n == 1) {
        ans = m;
    }
    else {
        ans = m + multiply(m, n - 1);
    }
    printf("%d\n", ans);
    return ans;
}

int main()
{
    printf("%d\n", multiply(6, 3));
    return 0;
}
```

```

5. #include<stdio.h>
void change(int *s[], int n) {
    int i, j, t;
    for (i = 0; i < 4; i++) {
        for (j = i; j < 4; j++) {
            t = *(s[j] + i);
            *(s[j] + i) = *(s[i] + j);
            *(s[i] + j) = t;
        }
    }
}
int main()
{
    int a[4][4] = {{1, 2, 3, 4}, {12, 13, 14, 5},
        {11, 16, 15, 6}, {10, 9, 8, 7}};
    int i, j, *p[4];
    for (i = 0; i < 4; i++) {
        p[i] = a[i];
    }
    change(p, 4);
    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            printf("%2d%c", *(p[i] + j), j < 3 ? ' ' : '\n');
        }
    }
    return 0;
}

```

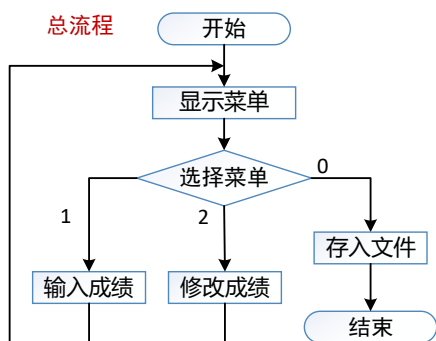
```

6. #include<stdio.h>
void silly1(int x) {
    int y;
    y = x + 2;
    x = x * 2;
}
void silly2(int *x) {
    int y;
    y = *x + 2;
    *x = 2 * *x;
}
int main()
{
    int x = 10, y = 11;
    silly1(x); printf("x=%d\n", x);
    silly1(y); printf("y=%d\n", y);
    silly2(&x); printf("x=%d\n", x);
    silly2(&y); printf("y=%d\n", y);
    return 0;
}

```

四、画流程图题

使用结构体数组 `st[N]` 存储学生的学号、姓名和成绩，编写“成绩录入程序”。该程序的总体流程如下图所示。



该程序包含“输入成绩”和“修改成绩”两个模块，模块的功能描述如下：

- 1) 输入成绩：循环 N 次，向结构体数组 `st` 中输入 N 个学生的学号、姓名和成绩。
 - 2) 修改成绩：输入一个学号，在数组 `st` 中查找该学号，如果在 `st[i]` 中找到该学号，则输入成绩并修改 `st[i]` 中的成绩，然后结束当前模块；如果没有在数组中找到该学号，则提示查询失败，然后结束当前模块。
- 请分别绘制上述两个模块的流程图。

综合测试题 (3)

一、 单选题

1. 以下选项中, 正确的标识符是 ()。
A. long B. _SUM C. f(x) D. 2x
2. 若变量 c 为 char 类型, 能正确判断出 c 为小写字母的表达式是 ()。
A. 'a'<=c<='z' B. (c>='a')||(c<='z')
C. ('a'<=c)and('z'>=c) D. (c>='a')&&(c<='z')
3. 若有定义: char str[]="ABCDEF";, 则 sizeof(str) 的值为 ()。
A. 4 B. 5 C. 6 D. 7
4. 若球体半径定义为: double r;;, 则求该球体体积的正确表达式为 ()。
A. 4/3.0*3.14159*(r^3) B. 4*3.14159*r*r*r/3
C. 4/3*3.14159*pow(r,3) D. 4/3*3.14159*r*r*r
5. 若有定义: int a=3, b=2, c=1, z;;, 则表达式 z=a>b>c 的值为 ()。
A. 0 B. 1 C. 2 D. 3
6. 下列关于 return 语句的表述中 () 是正确的。
A. 在函数体内 return 语句至少要出现 1 次
B. 在函数体内 return 语句只能出现 1 次
C. 函数返回值的数据类型取决于 return 语句所带的表达式的数据类型
D. 在函数体内 return 语句可以出现 0 次或多次
7. 若有定义 int a[3][4]; 则对 a 数组元素不正确的引用是 ()。
A. a[0][2*1] B. a[1][3] C. a[0][4] D. a[4-2][0]
8. 若有: int x,y; scanf("x=%d,y=%d",&x,&y);, 则能够使得 x 和 y 的值分别为 3 和 4 的正确输入方式为 ()。
A. x=3 y=4 B. x=3,y=4 C. 3,4 D. 3 4
9. C 程序中使用条件分支语句 if~else 时, else 应与 () 组成配对关系。
A. 同一复合语句内部的 if B. 在其之前任意的 if
C. 在其之前未配对的最近的 if D. 首行位置相同的 if
10. 设有定义 int k=0; 则以下 k 值不是 1 的是 ()。
A. k++ B. k+=1 C. ++k D. k+1
11. 有数组定义和函数 fun 调用语句 int a[3][4]; fun(a);, 则在函数 fun 定义时, 对形参 array 的错误定义方式为 ()。
A. fun(int array[][4]) B. fun(int array[3][4])
C. fun(int **array) D. fun(int (*array)[4])
12. 以下选项中, 操作数必须是整型或字符型的运算符是 ()。
A. ++ B. ! C. % D. /
13. 关于 C 语言程序, 以下叙述中正确的是 ()。

- A. main 函数必须位于所有其他函数之前
- B. 预处理命令属于一类特殊的 C 语言语句
- C. 优先级高的运算符优先计算
- D. C 语言的输入和输出功能只能通过函数调用才能实现

14. 以下程序的运行结果是 ()

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

- A. 2 B. 3 C. 4 D. 5

15. 若有定义:

```
struct student {
    int    num;
    char   name[16];
}stu, *p=&stu;
```

则能够正确输入 stu 中 num 和 name 成员的语句是 ()。

- A. scanf("%d%s", stu.num, &stu.name);
- B. scanf("%d%s", &stu.num, stu.name);
- C. scanf("%d%s", p->num, p->name);
- D. scanf("%d%s", &p.num, &p.name);

16. 若有语句 int *point, a=4; point=&a; 下面均代表地址的一组选项是 ()。

- A. a, point, *a B. &*a, &a, *point
C. &a, &*point, &point D. *&point, **&point, &a

17. 关于 break 语句和 continue 语句, 以下叙述中正确的是 ()。

- A. break 语句和 continue 语句仅可用于循环语句
- B. break 语句可直接退出多层循环
- C. continue 语句提前结束本次循环
- D. break 语句在退出循环时可携带一个返回值

18. 若有程序如下:

```
#include <stdio.h>
void swap(int* x, int* y){
    int *t;
    t=x, x=y, y=t;
}
int main(){
    int a=3, b=4;
    swap(&a, &b);
    printf("%d,%d", a, b);
    return 0;
}
```

则程序的输出为 ()。

- A. 3,3 B. 3,4 C. 4,4 D. 4,3

19. 若有定义: char str[8]="Hello", *p=str;, 则 strlen(p) 的值是 ()。

- A. 5 B. 6 C. 8 D. 不确定

20. 已知 ch 是字符型变量, 下面不正确的赋值语句是 ()。

- A. ch='a+b' B. ch='\0' C. ch='7'+'9' D. ch=5+9

21. 以下程序的运行结果是 ()。

```
#include <stdio.h>
void fun(int *p, int n) {
    int i,t;
    for(i=0;i<n/2;i++){
        t=*(p+i);
        p[i]=p[n-1-i];
        *(p+n-1-i)=t;}
}
int main() {
    int i,a[10]={9,8,7,6,5,4,3,2,1,0};
    fun(a,10);
    printf("%3d",a[5]);
}
```

- A. 5 B. 6 C. 4 D. 7

22. 若有: int a[4][5], *p=*a;, 则以下选项中可以表示 a[0][3] 的是 ()。

- A. p[0][3] B. p[3] C. *(a+3) D. **a+3

23. 以下程序的运行结果是 ()

```
#include<stdio.h>
int DigitSum(int n){
    if(n/10 == 0){
        return n;
    } else {
        return DigitSum(n/10)+n%10;
    }
}
int main ()
{
    int number=1234;
    printf("%d",DigitSum(number));
    printf("\n");
    return 0;
}
```

- A. 1234 B. 4321 C. 10 D. 24

24. 若有程序片段如下:

```
int **p, i, j;
p = (int **)malloc(10*sizeof(int **));
for (i=0; i<10; i++) {
    *(p+i) = (int *)malloc(5*sizeof(int));
    for (j=0; j<5; j++) {
        *(* (p+i)+j) = i*10+j;
    }
}
```

则上述程序片段执行结束后, $(*(p[4]+2))/(*(p[2]+4))$ 的值是 ()。

- A. 0 B. 1 C. 2 D. 3

二、不定项选择题

1. 如果打开文件时选用的文件操作方式为 "wb+", 以下说法正确的是 ()。

- A. 要打开的文件是二进制文件。
- B. 要打开的文件必须存在。
- C. 要打开的文件可以不存在。
- D. 打开文件后可以读取数据。

2. 关于函数的定义, 以下选项中正确的有 ()。

- A. 函数定义具有原型声明的作用。
- B. 函数定义时, 若未指定返回值类型, 则缺省为 `int` 类型。
- C. 函数定义时, 若返回值类型为 `void` 类型, 则表示没有返回值。
- D. 函数定义时, 若函数名后的圆括号中写作 `void`, 则表示没有参数。

3. 以下选项中, 当指针 `p` 为空指针时, 其值为真的表达式有 ()。

- A. `p` B. `!p` C. `p==NULL` D. `p=='\0'`

4. 以下说法正确的是 ()。

- A. 函数中的每个自动变量只在函数被调用时存在, 在函数执行完毕退出时消失。
- B. 外部变量必须定义在所有函数之外, 且只能定义一次。
- C. 函数的形参通常不需要单独分配内存。
- D. 不同函数中具有相同名字的局部变量之间没有关联。

5. 下列关于结构体类型和结构体变量的说法中, 正确的是 ()。

- A. “结构体” 可将不同数据类型、但相互关联的一组数据, 组合成一个有机整体使用。
- B. 结构体类型中成员名, 不可以与程序中的变量同名。
- C. “结构体类型名” 和 “数据项” 的命名规则, 与变量名相同。
- D. 相同类型的结构体变量间可以相互赋值。

6. 关于 C 语言中的 `switch` 语句, 以下选项中正确的有 ()。

- A. `switch` 语句是一种多分支语句。
- B. `switch` 语句中可以没有 `default` 分支。
- C. 程序执行到下一个 `case` 时, 跳出 `switch` 语句。
- D. `switch` 后的表达式可以是整型、字符型或浮点型。

三、填空题

1. 定义 `int a=0, b=0, c=0;` 语句 `c=2>1?(a=1):(b=2);` 执行后, 表达式 `a+b+c` 的值是 (1)。

2. 有 100 个数字从小到大排列, 若使用二分法进行查找, 则最多需要比较 (2) 次。

3. 已知 `a` 是一个 `double` 型的正数, 写出一个赋值表达式, 在它执行后 `a` 四舍五入保留两位小数 (3)。(如 `a` 的值原本是 12.666666, 执行完后变为 12.670000)

4. 定义 `char a[]="abcdef";` 则语句 `printf("%s", a+2);` 的结果是 (4)。

5. 若希望一个函数返回多个不同类型的数值, 可以将返回值定义为 (5) 类型。

6. 程序段:

```

unsigned char x=100, y=200;
do {
    x = x+y, y = x-y, x = x-y;
} while (0);
printf ("%d %d\n", x, y);

```

运行后，输出结果应为：(6) (7)。

7. 程序段：int i,s=0; for (i=1; i<=100; i++) s+=i; 运行后，s 的值为(8)。

8. 设：int a[2][3] = { {1, 2, 3}, {4, 5, 6} }, 则：*(*(&a[0] + 1))的值为(9)，
*(&a[0][0] + 3)的值为(10)。

四、程序填空题

1. 在一组有序的数据中查找数据，若找到则输出数据已在数组中，否则插入该元素。

```

#include <stdio.h>
#define N 10
void insert(int a[],int n, int m, int x) {
    int j; //该函数将x插入在a[m]
    for( (1) ; j>=m ;j--) a[j+1]=a[j];
    (2) ;
}
int main() {
    int a[N+1]={N个升序初始化值},i,x,flag=0;
    scanf("%d",&x); //输入待查找的数x
    for(i=0; i<N ; i++) {
        if(a[i]==x) {
            (3) ; break;
        }
        else if( (4) ) {
            break;
        }
    }
    if(flag==1) {
        printf("x is in array.\n");
    }
    else if(i<=N) {
        (5) ;
    }
    return 0;
}

```

2. 完成程序，填上适当的语句，实现功能：将输入的大写字母转换为小写字母、小写字母转换为大写字母、其它字符不变，并最后输出。每空仅写一个表达式或语句。

```

#include <stdio.h>
int main()
{
    char c;
    c=getchar();
    switch((c>='A') + (c>='Z') + (c>='a') + (c>='z')) {
        case 1: (6) ; break;
        case 3: (7) ; break;
    }
}

```

```

    printf("%c",c);
}

```

3. 有如下结构体类型，完成函数，实现用冒泡法按 score 降序对结构体数组 r 的元素进行排序。

```

struct student {
    char name[20];
    double score;
};

void BubbleSort(struct student r[],int n) { //冒泡法排序
    int i,j;
    (8) ;
    for(i=0;i<n-1;i++)
        for ( (9) ; j++)
            if( (10) ) { //比较成绩
                (11)
            }
}

```

4. 完成程序，填上适当的语句，实现功能：输入整数 n 的值，逆序输出 n 的各位数字。例如：输入 3210，输出：0123。每空仅写一个表达式或语句。

```

#include <stdio.h>

int main() {
    int n ;
    scanf("%d", &n );
    do {
        printf("%d", (12) );
    }while ( (13) );
    return 0;
}

```

5. 完成程序，填上适当的语句，实现功能：将一组字符串从小到大排序后输出。

```

#include <stdio.h>
#include <string.h>

void sortstring( (14) ) {
    int i, j, k;
    for (i = 0; i < n-1; i++) {
        for (k = (15) , j = (16) ; j < n; j++)
            if ( strcmp(p[j], p[k]) < 0 ) {
                k = j;
            }
        if (k != i) {
            char *t;
            t = p[i]; p[i] = p[k]; p[k] = t;
        }
    }
}

int main() {
    char *name[5] = {"Li Bai", "Du Fu", "Bai Juyi", "Du Mu", "Lu You"};
    int i;
}

```

```

    sortstring(name, 5);
    for (i = 0; i < 5; i++) {
        printf("%s\n", (17) );
    }
    return 0;
}

```

6. 以下函数的功能是计算 GPA 并返回。其中输入 n 是课程数，数组 gp[] 是每门课程的绩点，数组 credit 是相应课程的学分。但程序中有三处错误。请写出错误语句的行号，并改正。

```

(1) void GPA(double gp[], int credit[], int n) {
(2)     double s;
(3)     int i=0, c=0;
(4)     while (i<n) {
(5)         s+=gp[i]*credit[i];
(6)         c+=credit[++i];
(7)     }
(8)     return(s/c);
(9) }

```

五、画流程图题

有数组 char txt[T] = "USTC1958has33000...", 其中仅包含英文字母和数字，现需要将所有字母单独存放在一个数组中，形如 char str[S] = "USTChas...", 所有数字单独存放在另一个数组中，形如 char num[N] = "195833000...", 请用流程图描述该算法。

综合测试题（4）

一、 单选题

1. 关于 C 语言程序下列错误的是（ ）
- A) 每个 C 程序都必须在某个位置包含一个 main() 函数
B) 每个 C 程序都是由函数和变量组成的
C) 每个 C 程序都从 main 函数的起点开始执行
D) 由于 main 函数不能被任何函数调用，所以它不能带参数
2. 下列 C 语言用户标识符中合法的是（ ）
- A) _sum B) 2year C) long D) Mr.Wang
3. 将两个 8 位有符号数运算后的结果按 8 位无符号数处理其值为 160，实际上它代表（ ）
- A) -95 B) -96 C) -128 D) -160

4. 逻辑运算符两侧运算对象的数据类型（ ）
- A. 只能是 0 或 1 B. 只能是 0 或非 0 正数
C. 只能是整型或字符型数据 D. 可以是任何类型的数据

5. 以下程序所表示的分段函数是（ ）

```
#include<stdio.h>
int main() {
    int x,y;
    scanf("%d",&x);
    y=x>=0?2*x+1:0;
    printf("x=%d:f(x)=%d",x,y);
    return 0;
}
```

- A) $f(x) = \begin{cases} 0 & (x \leq 0) \\ 2x + 1 & (x > 0) \end{cases}$ B) $f(x) = \begin{cases} 0 & (x \geq 0) \\ 2x + 1 & (x < 0) \end{cases}$
C) $f(x) = \begin{cases} 2x + 1 & (x < 0) \\ 0 & (x \geq 0) \end{cases}$ D) $f(x) = \begin{cases} 0 & (x < 0) \\ 2x + 1 & (x \geq 0) \end{cases}$

6. 设有语句：int a=2,b=3,c=4; float x=3.5,y=4.8;
则表达式 $!(a+b)+c-1 \& \& b+c/2$ 和表达式 $x+a\%3*(int)(x+y)\%2/4$ 的值分别为（ ）
- A) 0 和 3.50000 B) 1 和 3.50000 C) 0 和 4.50000 D) 1 和 4.50000

7. 执行下列程序后，变量 i 的值是（ ）

```
int i=10,b=1;
switch ( i ) {
    case 9: ++i;
    case 10: i*2;
    case 11: b=(i==b,i+3,i/3);
            break;
    default: i+=1;
}
```

- A) 20 B) 2 C) 11 D) 1

8. 以下程序的输出结果是 ()

```
#include<stdio.h>
int main() {
    int a,b;
    for(a=1,b=1;a<=100;a++) {
        if(b>=10) { break; }
        if(b%3==1) {
            b+=3;
            continue;
        }
    }
    printf("%d\n",a);
    return 0;
}
```

A) 101 B) 6 C) 15 D) 4

9. 在 C 语言程序中，有关函数的定义正确的是 ()

- A) 函数的定义可以嵌套，但函数的调用不可以嵌套
- B) 函数的定义不可以嵌套，但函数的调用可以嵌套
- C) 函数的定义和函数的调用均不可以嵌套
- D) 函数的定义和函数的均可以嵌套

10. 以下程序的正确运行结果是 ()

```
#include<stdio.h>
int f(int a) {
    int b = 0;
    static int c = 4;
    b++; c++;
    return(a + b + c);
}
int main() {
    int a = 2, i;
    for (i = 0; i < 3; i++) { printf("%4d", f(a)); }
    return 0;
}
```

A) 8 8 8 B) 8 11 14
C) 8 10 12 D) 8 9 10

11. 以下程序的运行结果是 ()

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

A) 2 B) 3 C) 4 D) 5

12. 如下代码中，要获得“103”，以下描述不正确的是 ()

```
struct ND {
    int id;
    struct ND *next;
} *p, *q, *r;
```

```

p=(struct ND*)malloc(sizeof(struct ND)); p->id=101;
q=(struct ND*)malloc(sizeof(struct ND)); q->id=102;
r=(struct ND*)malloc(sizeof(struct ND)); r->id=103;
p->next=q;
q->next=r;
A) r->id                B) r->next->id
C) q->next->id          D) p->next->next->id

```

13. 以下代码的输出结果是 ()

```

int a[2][3]={1,2,3,4,5,6};
int (*p)[3]=&a[0];
printf("%d",(*++p)[1]);
p=a;
printf("%d",(*p)[1]);
A) 2,2                B) 2,5                C) 4,2                D) 5,2

```

14. int mystrlen(char *s) {

```

    int n;
    for (n = 0; *s != '\0'; s++)
        n++;
    return n;
}

```

int main()

```

{
    char s[10]="USTC";
    char *p1="USTC";
    char *p2=p1;
    ***
}

```

在主函数中***处调用 mystrlen 函数的错误语句是 ()

A) mystrlen(s); B) mystrlen(&s[0]);
 C) mystrlen(p1); D) mystrlen(*p2);

15. 已知 char x[]="hello", y[]={'h','e','l','l','o'}; 则关于两个数组长度的正确描述是 ()

A)相同 B)x 大于 y C)x 小于 y D)以上答案都不对

16. 已知学生记录及变量的定义如下

```

struct student {
    int no;
    char name[20];
    char gender;
    struct{int year,month,day;}birth;
}

```

struct student s,*ps;

ps=&s;

以下能给 s 中的 year 成员赋值 2005 的语句是 ()

A) s.year=2005; B) ps.year=2005;
 C) ps->year=2005; D) s.birth.year=2005;

17. 当运行时输入: abcd\$abcde, 下面程序的运行结果是 ()

```
#include<stdio.h>
int main() {
while(putchar(getchar())!='$');
printf("end");
}
```

A) abcd\$abcde B) abcdend C) abcd\$end D) abcd\$abcdeend

18. 下列语句中, 将 p 定义成一个指针型变量的是 ()。

A) double *p[5]; B) double (*p)[5];
C) double *p(5); D) double *p();

19. 以下程序的运行结果是 ()

```
#include <stdio.h>
void fun(int x) {
    if(x/2>0) {
        fun(x/2-2);
        printf("%d ",x);
    }
}
int main() {
    fun(20);
    printf("\n");
    return 0;
}
```

A) 20 8 2 -1 B) 1 2 8 C) 8 D) 2 8 20

20. 以下代码运行后, i,j,k,m,n 的输出结果为 ()

```
#include <string.h>
int main()
{
    char *ps="0123456789";
    char buffer[]="Hello";
    int i=sizeof(ps);
    int j=sizeof(*ps);
    int k=strlen(ps);
    int m=strlen(buffer);
    int n=sizeof(buffer);
    printf("%d %d %d %d %d\n",i,j,k,m,n);
    return 0;
}
```

A) 8 1 10 5 6 B) 8 4 10 5 6 C) 1 1 11 6 5 D) 8 1 10 5 5

21. 以下说法错误的是 ()

A) 指针是一种保存变量地址的变量。
B) 一般情况下, 同其它类型的变量一样, 指针也可以初始化。
C) 对指针有意义的初始化只能是表示地址的表达式。
D) 指针若赋值 NULL 则指向 0x0 这个地址。

22. 以下程序的运行结果是()

```
#include <stdio.h>
void fun(int *p, int n) {
    int i,t;
    for(i=0;i<n/2;i++) {
        t=*(p+i);
        p[i]=p[n-1-i];
        *(p+n-1-i)=t;
    }
}
int main() {
    int i,a[10]={9,8,7,6,5,4,3,2,1,0};
    fun(a,10);
    printf("%3d",a[5]);
}
```

- A) 5 B) 6 C) 4 D) 7

23. 以下程序的运行结果是 ()

```
struct {
    int id;
    char name[15];
}stu[4]={2101,"Darkness",2102,"Gorgeous",2103,"Light",2104,"Tread"},*p=stu;
p++;
printf("%c\n",++p->name[1]);
```

- A) E B) o C) p D) L

24. 若有函数定义如下:

```
int func(int n) {
    if (n>0)
        return n+func(n-1);
    return 0;
}
```

则 func(10) 的值为 ()。

- A. 0 B. 10 C. 45 D. 55

二、不定项选择题

1. 以下表达式的值是整型的有 ()

- A: sizeof(double) B: 3.5-0.5 C: 'x' D: 3.5>0.5

2. 设 x、y 和 z 是 int 型变量, 且 x=3,y=4,z=5,则下面表达式中值为 0 的是 ()

- A) 'x'&&'y' B) x>=y; C) x||y+z&& y-z D) !((x<y)&&!z||1)

3. 以下关于编译预处理的叙述中正确的是 ()

- A) 预处理命令行必须以#开始
B) 一条有效的预处理命令必须单独占据一行
C) 预处理命令行只能位于源程序中所有语句之前
D) 预处理命令不是 C 语言本身的组成部分

4. 若有说明语句如下:

```
char a[]="It is mine";  
char *p=a;
```

则以下正确的叙述是 ()

- A) a+2 表示的是字符't'所在存储单元的地址
- B) p 指向另外的字符串时, 字符串的长度不受限制
- C) *(p+i)等价于 p[i]
- D) a 中只能存放 10 个字符

5. 以下说法正确的是()

- A) 在函数之外定义的变量是全局变量
- B) 全局变量可以被本文件中的其他函数访问
- C) 局部变量仅仅在其所在的函数内部范围内有效
- D) 静态 (static) 变量的生存期贯穿于整个程序的运行期间

三、填空题

1. 已定义 float x = 213.82631; 语句 printf("%-4.2f\n",x); 的输出结果是 (1) _____
2. 若有以下定义: char c = '\X41'; 则变量 C 中包含的字符个数为 (2) _____
3. 有定义 char str[]="nUSTC\n2021"; 则表达式 sizeof(str)/sizeof(str[0]) 的值是: (3) _____
4. 若有以下定义: char array[2]="0", *p=array; 则表达式 (*p++) == '\0' 的值为: (4) _____
5. 程序中有下列程序语句:

```
unsigned char x=100, y=200;  
do{  
    x = x+y, y = x-y, x = x-y;  
}while (0);  
printf ("%d %d\n", x, y);
```

输出结果应为: (5) _____ (6) _____

6. 设有定义 char x, y; 请写出描述 "x, y 同时为小写字母或者同时为大写字母" 的表达式: (7) _____。
7. 定义 int i=100; 执行语句 while(i-->0) { if (!i) break;} 后, i 的值为 (8) _____。
8. 函数调用语句 fun(rec1, rec2+rec3, (rec4, rec5)); 含有的实参个数是 (9) _____。
9. 若有 int i=1, a[10][5]={1,2,3,4,5,6,7,8,9,10}, (*p)[5]=a; 则表达式 (*(p+i)+5) 的值为 (10) _____

四、程序填空题

1. 统计给定数组 a 中素数的个数并输出

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

```
int prinum(int *a)
```

```
{  
    int count=0, i, j, k;  
    for (i=0; i<10; i++) {  
        for (j=2; j<=a[i]-1; j++) {  
            if (_____(1)_____) {  
                _____(2)_____;  
            }  
        }  
        if (_____(3)_____) {
```

```

        count++;
    }
}
return count;
}
int main()
{
    int a[10]={11,3,50,17,81,9,10,101,111,12};
    printf("prime numbers are: %d \n", prinum(____(4)____));
}

```

2. 统计字符串中字母的个数，请填空。

```

int main() {
    char str[50];
    int i,____(5)____;
    scanf("%s",____(6)____);
    for(i=0;____(7)____;i++)
        if(____(8)____) j++;
    printf("j=%d\n",j);
}

```

3. 数组 a 中存放 N 个由小到大排列的有序整数。把从键盘输入的整数 m 插入到数组 a 中，使插入后的数组 a 仍然有序，请填空。

```

#define N 6
int main() {
    int i,j,m;
    int a[____(9)____]={10,20,30,40,50,60};
    scanf("%d",&m);
    for(j=0;j<N;j++)
        if(____(10)____<a[j]) break;
    for(i=N;i>j;i--)
        a[i]=a[____(11)____];
    a[j]=____(12)____;
    for(i=0;i<N+1;i++)
        printf("%d",a[i]);
}

```

4. 下列函数 tax 根据收入金额 salary (≥ 0) 对应不同税率计算应缴税额，并返回应缴税额。税率计算公式 $f(x)$ 如下：

$$f(x) = \begin{cases} 0, & x < 1000 \\ 5\%, & 1000 \leq x < 3500 \\ 10\%, & 3500 \leq x < 5000 \\ 15\%, & x \geq 5000 \end{cases}$$

```

float tax(int salary){
    switch (____(13)____){
        case 0:
        case 1:
            return 0;
        case 2:

```

```

        case 3:
        case 4:
        case 5:
        case 6:
            _____(14)_____;
        case 7:
        case 8:
        case 9:
            return salary*0.1;
        default:
            _____(15)_____;
    };
}

```

5. 折半查找算法。已知数组中的元素按照从小到大排列，本函数使用折半查找算法从数组中查找指定数字。若找到该数字，则返回该元素的下标；若未找到该数字，则返回 -1。

```

#include <stdio.h>

#define N 11

int binary_search (int *a, int n, int key)
{
    int low=0, mid, high=n-1;
    while (low <= high) {
        mid = (low + high) / 2;
        if (key > a[mid]) {
            low = _____(16)_____;
        } else if (key == a[mid]) {
            return _____(17)_____;
        } else {
            high = _____(18)_____;
        }
    }
    return _____(19)_____;
}

int main()
{
    int a[N]={5, 13, 19, 21, 37, 56, 64, 75, 80, 88, 92};
    int key, index;
    printf("Input key: \n");
    scanf("%d", &key);
    index = binary_search(_____(20)_____);
    if (index < 0) {
        printf("The key is not found!\n");
    } else {
        printf("index=%d, key=%d\n", index, key);
    }
    return 0;
}

```

五、画流程图题

模块功能需求：从键盘输入一行字符串，统计以空格作为分隔的单词个数。请用流程图描述实现该模块功能的算法。假设以下变量已定义：

`char c, str[200];` // 字符数组 `str` 可以用于存储键盘输入的字符串

`int i, num=0, word=0;` // `num` 作为单词计数器，`word` 可以作为新单词开始的标志量。

综合测试（1）参考答案

一、单选题

1. C

解释：ANSI C 没有幂运算符，计算 x^n 需要用幂运算函数 `pow(x,n)`

2. A

解释：C 语言中每个操作数只能和一个运算符结合，并且要按优先级从高到低结合，`10<a<15` 本意想表达的是 `a` 操作数比 `10` 大且比 `15` 小，但在 C 语言中不能写成 `10<a<15`，因为这样的表达式会首先计算 `10<a`，得到的结果是逻辑值 `0` 或 `1`，之后再用 `0` 或 `1` 与 `15` 比较，显然不合题意，因此应该按 C 的语法先写出两个关系表达式 `10<a` 或 `a>10` 以及 `a<15` 或 `15>a`，再写成逻辑表达式 `a>10&&a<15`。B 和 D 的结果与 C 相同。

3. C

解释：空语句只是什么都不做，并不是错误语句

4. B

解释：`3/2` 是整数除法，结果是 `double (x + 1) = 2.0`

5. A

解释：逗号表达式虽然只要右边的值，但先算左边的表达式，`p+=2` 相当于移到了 `a[2]`，`*p` 的值已经是 `3` 了，只能用后缀++

6. C

解释：由于有 `()`，`*` 优先与 `ptr` 结合，`ptr` 是指针，指向有 `10` 个元素的数组

7. B

解释：`double` 型数据输入用 `%lf`

8. B

解释：字符串的结尾 `'\0'` 也占一个字节

9. C

解释：A. 函数返回值为整型时，定义时允许缺省，但声明时必须写；

比如：对于函数 `f` 的定义 `f(int x) {return x;}` 声明必须要写 `int f(int x);`

递归定义是指在一个函数里面再定义自己，这是 C 不允许的

10. B

解释：`p` 指向 `a` 的地址，`*p` 是取这个地址里的内容

二、填空题

1. 全局、局部

2. 5

3. 1, 0

4. 0

5. 'a' 或 97

6. 0

7. 20

8. 6

三、阅读程序后写出运行结果

1. 运行结果:

```
4
7
10
10
4
```

2. 运行结果

```
17154
```

3. 运行结果

```
###1###1
###9###1
##17
```

4. 运行结果

```
###0###2###4###4###8
```

注:#指空格

5. 运行结果

```
image
mage
age
ami
```

6. 运行结果

```
1,3
3,1
1,3
```

7. 运行结果

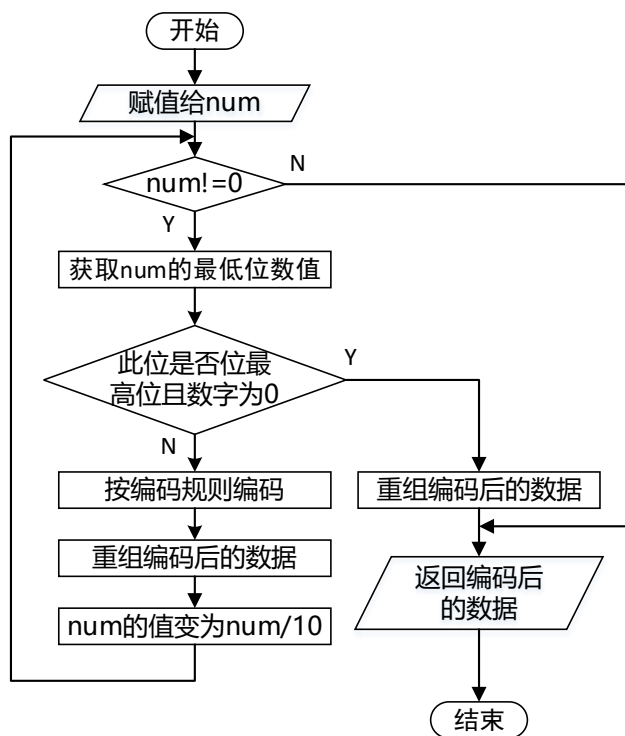
```
10 14
28 32
84
```

四、程序填空题

- (1) double a[], int n 或 double a[5], int n 或 double *a, int n
- (2) t*x
- (3) y
- (4) x[i][i]
- (5) x[i-1][j]+x[i-1][j-1]

五、画流程图题

移位编码算法流程图



综合测试（2）参考答案

一、单选题

1. A 2. C 3. B 4. D 5. B
6. A 7. B 8. C 9. C 10. D

二、填空题

1. 执行什么样的操作 或 执行的运算
2. 保留字
3. $T = 4 * t * t / (t + 2) - 20$ 或 $T = 4 * \text{pow}(t, 2) / (t + 2) - 20$
4. 1
5. 1
6. n 的值是 2, m 的值是 8, p 的值是 6
7. x 的值是 21, y 的值是 1, z 的值是 23
8. 不是, 表达式非法。递增运算符不能作用于 $(n * m)$ 这样的表达式。

三、阅读程序后写出运行结果

1. 运行结果:

```
i=2, x=6  
i=3, x=3  
i=4, x=8
```

2. 运行结果

```
x=3  
x=6  
x=14
```

3. 运行结果

```
1 5 3 2 4  
1 2 3 5 4  
1 2 3 5 4  
1 2 3 4 5  
1 2 3 4 5
```

4. 运行结果

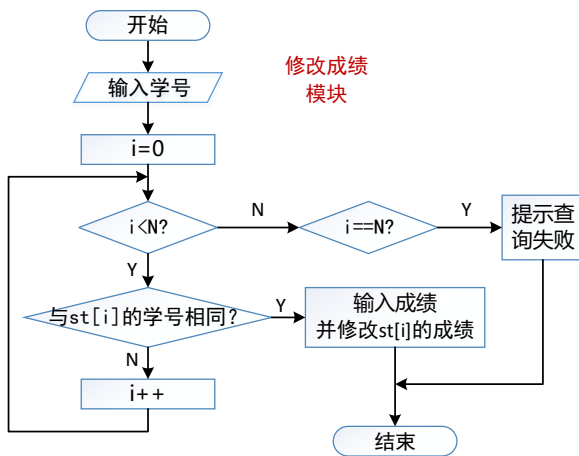
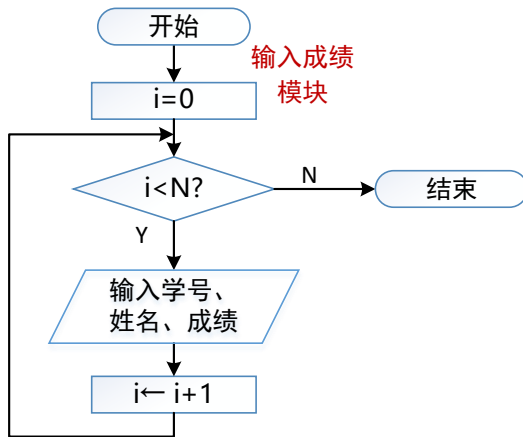
```
6  
12  
18  
18
```

5. 运行结果

1 12 11 10
2 13 16 9
3 14 15 8
4 5 6 7

x=10
y=11
x=20
y=22

6. 运行结果
四、画流程图题



综合测试（3）参考答案

一、单选题

- | | | |
|------|-------|-------|
| 1. B | 9. C | 17. C |
| 2. D | 10. D | 18. B |
| 3. D | 11. C | 19. A |
| 4. B | 12. C | 20. A |
| 5. A | 13. D | 21. A |
| 6. D | 14. B | 22. B |
| 7. C | 15. B | 23. C |
| 8. B | 16. C | 24. B |

二、不定项选择题

- | | | |
|---------|--------|--------|
| 1. ACD | 3. BCD | 5. ACD |
| 2. ABCD | 4. ABD | 6. AB |

三、填空题

1. (1) 2
2. (2) 7
3. (3) a=(int) (a*100+0.5)/100.0
4. (4) cdef
5. (5) 结构体（或结构）
6. (6) 200 (7) 100
7. (8) 5050
8. (9) 4 (10) 4

四、程序填空题

1.
(1) `j=n-1`
(2) `a[m]=x`
(3) `flag=1` 或 `flag+=1` 或 `flag=flag+1`
(4) `a[i]>x`
(5) `insert(a,N,i,x)`
2.
(6) `c=c-'A'+'a'` 或 `c=c+32`
(7) `c=c-'a'+'A'` 或 `c=c-32`
3.
(8) `struct student t`（或其它变量名）
(9) `j=0; j<n-i-1`
(10) `r[j].score<r[j+1].score` 或反过来
(11) `t=r[j],r[j]=r[j+1],r[j+1]=t;`（分号逗号都可以）

4.

(12) $n \% 10$

(13) $n \neq 10$ 或 $(n = n/10) \neq 0$ 或 $n = n / 10$ 或 $(n \neq 10) \neq 0$

5.

(14) `char **p, int n` 或 `char *p[5], int n`

(15) `i`

(16) `i+1`

(17) `name[i]` 或 `*(name+i)`

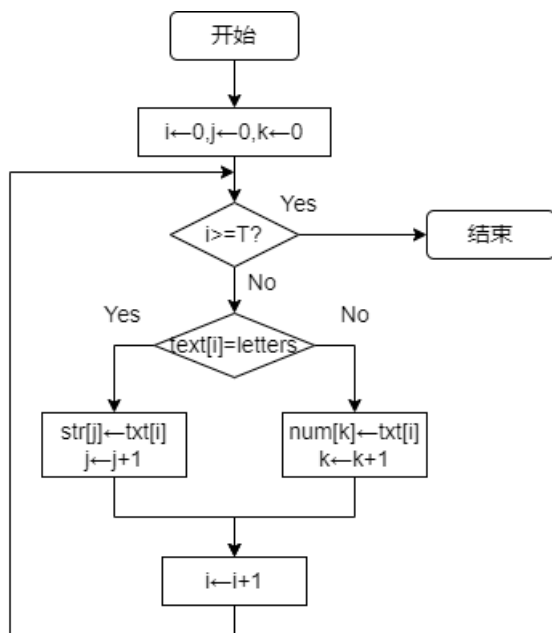
6.

(18) (1): `void` 替换为 `double`

(19) (2): `double s=0;`

(20) (6): `c+=credit[i++];`

五、画流程图题



综合测试（4）参考答案

一、单选题

- | | | |
|------|-------|-------|
| 1. D | 9. B | 17. C |
| 2. A | 10. D | 18. B |
| 3. B | 11. B | 19. D |
| 4. D | 12. B | 20. A |
| 5. D | 13. D | 21. C |
| 6. B | 14. D | 22. A |
| 7. B | 15. B | 23. C |
| 8. D | 16. D | 24. D |

二、不定项选择题

- | | | |
|--------|--------|---------|
| 1. ACD | 3. ABD | 5. ABCD |
| 2. BD | 4. BC | |

三、填空题

- (1) 213.83
- (2) 1
- (3) 11
- (4) 0
- (5) 200 (6) 100
- (7) $(x \geq 'A' \ \&\& \ y \geq 'A' \ \&\& \ x \leq 'Z' \ \&\& \ y \leq 'Z' \ || \ x \geq 'a' \ \&\& \ y \geq 'a' \ \&\& \ x \leq 'z' \ \&\& \ y \leq 'z')$
- (8) 0
- (9) 3
- (10) 0

四、程序填空题

- (1) $a[i] \% j == 0$
 - (2) break
 - (3) $j == a[i]$
 - (4) a
- (5) $j = 0$
 - (6) str
 - (7) $str[i] != '\0'$ 或 $i < strlen(str)$
 - (8) $str[i] \geq 'A' \ \&\& \ str[i] \leq 'Z' || str[i] \geq 'a' \ \&\& \ str[i] \leq 'z'$
- (9) $N+1$

(10) m
 (11) i-1
 (12) m
 4.
 (13) salary/500
 (14) return salary*0.05 ;
 (15) return salary*0.15 ;
 5.
 (16) mid+1
 (17) mid 或 (mid)
 (18) mid-1
 (19) -1
 (20) a, N, key 或 a, 11, key

五、画流程图题

