



7、下面定义二维数组不正确的选项是 ( )

A) `int b[2][ ]={{4,6},{5,7}};`

B) `int b[ ][2]={5,6,2,3};`

C) `int b[2][2]={{4},{6}};`

D) `int b[2][2]={{4},5,6};`

8、定义 `char a=127, b;` 计算 `b=a++;` 后, `a`、`b` 的值分别为 ( )

A) `a=127; b=127;`    B) `a=128; b=128;`

C) `a=-127; b=128;`    D) `int b[2][2]={{4},5,6};`

9、变量的有效范围与其定义位置有关, ( ), 其作用域在整个源程序文件中都有效。

A) 在第一个函数中定义的变量

B) 在定义第一个函数之前所定义的变量

C) 在主函数中定义的变量

D) 在函数中定义的静态变量

10、当 `c` 的值不为 0 时, 下列选项中能将 `c` 的值赋给变量 `a`、`b` 的是 ( )

A) `c=b=a;`    B) `(a=c) || (b=c);`

C) `(a=c) && (b=c);`    D) `a=c=b;`

11、以下定义中, `p` 不是指针变量的是 ( )。

A) int \*p;      B) int (\*p)[2];      C) int \*\*p;      D) int \*p[2]

12、已知 char b[5], \*p=b; 则正确的赋值语句是 ( )。

A) b="abcd";      B) \*b="abcd";

C) p="abcd";      D) \*p="abcd";

13、以下程序 ( )。

```
int x = -1;

do{
    x = x*x;
}while (!x);
```

A) 是死循环      B) 循环执行 2 次

C) 循环执行 1 次      D) 循环执行 0 次

14、以下不同进制数中值最大的是 ( )。

A) (1101111)<sub>2</sub>      B) (65)<sub>8</sub>      C) (54)<sub>10</sub>      D) (43)<sub>16</sub>

15、C 语言程序从 ( ) 开始执行。

A) 程序中第一条可执行语句      B) 程序中第一个函数

C) 程序中的 main 函数      D) 包含文件中的第一个函数

16、若在定义函数时未说明函数的类型，则缺省的函数类型是 ( )。

A) char      B) int      C) void      D) double

17、以下运算符是按优先级从高到低排列的是 ( )。

A) + && <=      B) <= + &&      C) + <= &&      D) <= && +

18、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]`

19、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。

20、定义 `FILE *fp`；下列语句中打开文件后不能修改其中内容的是 ( )。

A) `fp=fopen("myfile", "w")`；

B) `fp=fopen("myfile", "rb")`；

C) `fp=fopen("myfile", "ab")`；

D) `fp=fopen("myfile", "r+")`；

## 二、填空 (共 20 分, 每空 1 分)

1、定义 `char a='B'`；语句 `printf("c", a-'A'+'a')`；

的执行结果是\_\_\_\_\_。

2、设有定义：`int a[2][3]={1,2,3,4,5,6},(*p)[3]=a`；则 `*(a+1)` 的值为

\_\_\_\_\_，`*(a+1)` 的值为\_\_\_\_\_，`*p[1]` 的值为\_\_\_\_\_，

(\*p)[1] 值为\_\_\_\_\_。

3、以下程序在降序数组中进行二分查找，请补充完成该程序。

```
#include<stdio.h>

int binarysearch(_____) {
    int low=_____, mid, high=_____;
    while(low>=high) {
        mid=(low+high)/2;
        if(key<r[mid]) high=mid+1;
        else if(key==r[mid]) return(mid);
        else _____;
    }
    _____;
}

main() {
    int a[11]={92, 88, 80, 75, 64, 56, 37, 21, 19, 13, 5};
    int i, k;
    scanf("%d", &k);
    i=binarysearch(a, 11, k);
    if(i!=-1) printf("a[%d]=%d\n", i, a[i])
    else printf("not found! \n");
}
```

4、以下程序中，主函数接受一个字符串到数组 s，调用函数 reverse 将数组中的字符串逆序存放，再输出。请补充完成该程序。

```
#include <stdio.h>

void reverse(_____str) {
    int len, j;
    char c;
    for (len=0; _____; len++);
    for (j=0; j<_____; j++) {
        c = str[j];
        str[j] = str[len-1-j];
        str[len-1-j] = c;
    }
}

int main () {
    char s[256];
    scanf (_____, s);
    reverse (_____);
    puts (s);
    return 0;
}
```

5、以下程序中，主函数循环输入数值 n (直至  $n \leq 0$  结束)，调用函数 printstar 打印 n 次\*，在函数里计数并显示这是第几次打印。请补充完成该程序。

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

```
void _____
```

```
int main () {
```

```
    int n=1;
```

```
    do {
```

```
        printfstar (n) ;
```

```
        scanf ("%d", _____) ;
```

```
    } _____
```

```
}
```

```
void printfstar (int a) {
```

```
    _____
```

```
    int i;
```

```
    printf("这是第%2d 次打印: ", n++);
```

```
    for (i=0; _____ i++) printf ("*");
```

```
    printf ("\n");
```

```
}
```

阅读程序,

写出程序运行结果 (共 25 分, 每题 5 分)

### 第 1 题

```
#include <stdio.h>

#define N 3

#define M N+2

#define PLUS(a, b) a+(b)

Int main() {

    Int x=3, y=3;

    printf("%d, %d\n", M, N);

    printf("%d\n", M*N);

    printf("%d\n", M+N);

    printf("%d\n", PLUS(M, N));

    y *=PLUS(x, y+1);

    printf("%d\n", y~~

}
```



## 第 2 题

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

```
int main () {  
    int i, j, n;  
    for(i=0; i<=2; i++) {  
        for(j=0; j<=2; j++) {  
            n=i+j;  
            if(n<=1) break;  
  
            printf("i+j=%d\n", n);  
            if (n! =3) continue;  
            printf("%d: %d, %d\n", n, 1, j);  
        }  
    }  
}
```

## 第 3 题

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

```
void main () {  
    char s[10]="abcde", i;  
    for(i=0; i<5; i++) {  
        switch(i) {  
            case 2: printf("B, ");  
            case 3: printf("%c", s[i]); break;  
  
            default: printf("%d: %c\n", i, s[i]);  
            case 4: printf("%c", s[i]); break;  
        }  
    }  
}
```

#### 第4题

```
#include <stdio.h>

void recurse(int *a, int k) {

    if(a[k] ==) return;

    recurse(a, k*2);

    printf("%d\n", a[k]);

}

main () {

    int b[15]={0, 1, 2,3,4,5};

    recurse(b, 1);

}
```

#### 第5题

```
#include <stdio.h>

void swap1(int *a, int *b) {

    int *t;

    t=a; a=b; b=t;

}

void swap2(int *a, int *b) {

    int t;

    t=*a; a=*b; *b=t;

}

main () {

    int a=3, b=4, *pa=&a, *pb=&b;

    swap1(&a,&b);

    printf("%d, %d\n", a, b);

    swap2(pa, pb);

    printf("%d, %d\n", a, b);

    swap3(pa, &pb);

    printf("%d, %d, %d, %d\n", a, b, *pa, *pb);

}
```

## 编程应用题

(共 35 分，空间不够可写在答题纸背面并注明)

### 第 1 题(7 分)

定义函数 num，通过形参传入双精度浮点数数组 a(数组中所有元素值不重复且已单调增排序)，返回最接近且小于等于所有元素平均值的元素值。

如： $a[5]=\{0.1, 0.2, 0.3, 0.4, 0.5\}$  返回 0.3

### 第 2 题(8 分)

规定一串密码由 8-15 个字符组成，其中至少有一个大写字母，至少一个小写字母，至少一个 0-9 的数字字符，定义一个函数，判断形参 s 的长度和字符是否符合上述要求，长度不符输出 -1，字符不符输出 0，全部符合输出 1.

函数原型说明：`int isPassword(char *s)`；其中长度计算不允许调用库函数。

### 第3题(20分)

编程实现从键盘循环输入学生成绩信息(如: pb16001091 林子涵 87) 并按学号顺序插入链表和写入文件中, 部分程序见题后, 补充完整的程序, 要求:

- 1 在 main 函数中每输入一个学生成绩信息创建一个新结点并调用 SortInsert 函数将该结点插入链表中, 结束循环(输入学号为"0")后将链表的每个结点作为一行(学号, 姓名, 成绩之间以空格分隔)写入文件"d:\temp\grade.dat"中
- 2 自定义字符串比较函数 strcmp, 按 ASCII 码顺序比较字符串的大小;
- 3 定义 SortInsert 函数, 该函数调用字符串比较函数 strcmp, 将新结点按学号递增的顺序插入链表并返回链表头指针。

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

```
#include <stdlib.h>
```

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

```
struct student {
```

```
    char num [11];
```

```
    char name [20];
```

```
    int score;
```

```
    struct student *next;
```

```
};
```

```
int strcmp(char *a, char *b);
```

```
struct student *SortInsert(struct student *head, struct student *ps);
```

```
main() { //无需抄写 main() 以上的=语句
```

```
    struct student s, *ps=NULL, *head =NULL;
```

```
    FILE *fp
```

```
    scanf("%s", s.num); }
```