

## C 语言程序设计（上）复习题

### 一、单项选择题（每题 2 分，共 40 分）

1. 有变量定义 `int a; float c;` 下面输入语句不能正确赋值的是( )。  
A. `scanf("%d,%f", &a, &c);`  
B. `scanf("x=%d, c=%f", &a, &c);`  
C. `scanf("%d%f", &a, &c);`  
D. `scanf("%f%d", a, c);`
2. 已知 `int k=1; for(k=1; k++; if(k) break;` `for` 语句的循环次数是( )。  
A. 0  
B. 1  
C. 2  
D. 无限次
3. 表达式 `!(x>0||y>0)` 等价于\_\_\_\_\_。  
A. `!x>0||!y>0`  
B. `!(x>0)||!(y>0)`  
C. `!x>0&&!y>0`  
D. `!(x>0)&&!(y>0)`
4. 二维数组 `a` 有 `M` 行 `N` 列，则在 `a[i][j]` 之前的元素个数为\_\_\_\_\_。  
A) `j*M+i`    B) `i*N+j`    C) `i*N+j-1`    D) `j*M+i-1`
5. 若已定义 `x` 为 `int` 类型, `y` 为 `float` 型，则表达式: `x=2, x+5, y=x+1/4` 的值是\_\_\_\_\_。  
A) 2.0    B) 2.25    C) 7    D) 7.25
6. 把字符串“China bit”输入到字符数组 `a` 中的正确操作是\_\_\_\_\_。  
A) `scanf("%s", a)`    B) `gets(a)`    C) `scanf("%s", &a)`    D) `getchar(a)`
7. 为判断 `char` 型变量 `m` 是否是数字，可以使用下列表达式\_\_\_\_\_进行判断。  
A) `0<=m && m<=9`    B) `'0'<=m && m<='9'`  
C) `"0"<=m && m<="9"`    D) A) `0<=m || m<=9`
8. 有 `int x, y;` 则以下 `for` 语句:  
`for(x=0, y=10; (y>0)&&(x<4); x++, y--);`  
其循环执行的次数是\_\_\_\_\_。  
A) 无限次    B) 循环次数不定    C) 循环执行 3 次    D) 循环执行 4 次
10. 下列说法中正确的是\_\_\_\_\_。  
A) `break` 语句用在 `switch` 语句中，而 `continue` 用在循环语句中  
B) `break` 语句用在循环语句中，而 `continue` 用在 `switch` 语句中  
C) `break` 能结束循环，而 `continue` 只能结束本次循环开始下次循环

D) continue 能结束循环，而 break 只能结束本次循环

11. 以下标识符中, 可以作为合法的C用户定义标识符的是 ( )。

A: 6answer

B: b/et

C: unsigned

D: \_if

12. 下面有关for 循环的正确描述是 ( )。

A: for 循环必须用于循环次数已经确定的情况

B: for 循环是先执行循环体语句，后判断表达式

C: for 循环中，不能用break 语句跳出循环体

D: for 循环的循环体语句中，可以包含多条语句，但必须用花括号括起来

13 下列定义数组的语句中正确的是 ( )。

A: #define size 10

char str1[size],str2[size+2];

B: char str[];

C: int num['10'];

D: int n=5; int a[n][n+2];

14.C 语言中的标识符只能由字母，数字和下划线三种字符组成，且第一个字符 ( )。

A: 必须为字母

B: 必须为下划线

C: 必须为字母或下划线

D: 可以是字母，数字和下划线中任一字符

15. 温度华氏和摄氏的关系是:  $C = 5/9 \times (F - 32)$ 。已知: float C,F; 由华氏求摄氏的正确的

赋值表达式是\_\_\_\_\_。

A)  $C = 5/9(F - 32)$

B)  $C = 5 * (F - 32) / 9$

C)  $C = 5/9 * (F - 32)$

D) 三个表达式都正确

16 已知 `int a[3][2] = {3, 2, 1}`; 则表达式“`a[0][0]/a[0][1]/a[0][2]`”的值是\_\_\_\_\_。

A) 0.166667

B) 1

C) 0

D) 错误的表达式

17. 对于强制类型转换，正确的说法是\_\_\_\_\_。

A) 对于进行强制类型转换的变量，按照转换后的类型进行保存

B) 对于进行强制类型转换的变量，变量本身的数据类型不变

C) 对于进行强制类型转换的变量，变量本身的数据类型和运算时的类型都改变

D) 对于进行强制类型转换的变量，变量本身的数据类型和运算时的类型都不改变

18. 使用语句 `scanf("x=%f,y=%f",&x,&y)`; 输入变量 x、y 的值 (□代表空格)，正确的输入是\_\_\_\_\_。

A) 1.25,2.4

B) 1.25□2.4

C) x=1.25,y=2.4

D) x=1.25□y=2.4

19.求取满足式  $1^2+2^2+3^2+ \dots +n^2 \leq 1000$  的  $n$ ，正确的语句是\_\_\_\_\_。

- A) `for(i=1,s=0; (s=s+i*i)<=1000; n=i++);`  
B) `for(i=1,s=0; (s=s+i*i)<=1000; n=++i);`  
C) `for(i=1,s=0; (s=s+++i*i)<=1000; n=i);`  
D) `for(i=1,s=0; (s=s+i*i++)<=1000; n=i);`

20.错误的转义字符是\_\_\_\_\_。

- A) `'\'`      B) `'\"'`      C) `'\81'`      D) `'\0'`

## 二、程序题（每题 20 分，共 60 分）

### 【程序题 1】打印图形题

输入顶行字符和图形的高。输出如下例（顶行字符为'A'，图形的高为 5）所示的图形，

```
A
BCD
EFGHI
JKLMNOP
QRSTUVWXYZ
```

测试用例：

输入：

A5

输出：

```
A
BCD
EFGHI
JKLMNOP
QRSTUVWXYZ
```

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

```
char check(char k)
```

```
{
```

```
    if (k>=65&& k<=90) return k;
```

```

        else return(k-26);

    }

int main()

{

    int i,j,n;char a;

    scanf("%c%d",&a,&n);

    for(i=1;i<=n;i++)

    {

        for(j=1;j<=n-i;j++)

        {

            printf(" ");

        }

        for(j=1;j<=2*i-1;j++)

        {

            printf("%c",check(a));

            a=a+1;

        }printf("\n");

    }

}

```

## 【程序题 2】“粘贴”字符串

编程，输入两个字符串 s1、s2 和 s1 中任意字符 k，在 s1 中的指定字符 k 的位置处插入字符串 s2 并输出。

例如 S1=“abcde”，S2=“12345”，输入 b，则输出“ab12345cde”，

若输入的字符不在 s1 中，则在 s1 的结尾处插入 s2。例如 S1="abcde",S2="12345",输入字符 k，则输出“abcde12345”。

测试用例：

输入：

abcde

12345

b

输出：

ab12345cde

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

```
int main()
```

```
{
```

```
    char a[50] = { '\0' }, b[50] = { '\0' }, out[50] = { '\0' };
```

```
    char c;
```

```
    int i;
```

```
    int setoff;
```

```
    int blen;
```

```
    gets(a);
```

```
    gets(b);
```

```
    c = getchar();
```

```
    for (i = 0; a[i]!='\0' && a[i]!=c; i++)
```

```
    {
```

```
        out[i] = a[i];
```

```
    }
```

```
    if (a[i] != '\0')
```

```
    {
```

```
        out[i] = a[i];
```

```
        setoff = i + 1;
```

```
    }
```

```
    else setoff = i;
```

```
    for (i = 0; b[i] != '\0'; i++)
```

```
    {
```

```
        out[i + setoff] = b[i];
```

```

    }

    blen = i;

    for (i = setoff; a[i] != '\0'; i++)
    {
        out[i + blen] = a[i];
    }

    out[i + blen] = '\0';

    printf("%s\n", out);

    return 0;
}

```

### 【程序题 3】寻找鞍点

编程，找出一个二维数组中的鞍点，即该位置上的元素是该行上的最大值，是该列上的最小值。二维数组也可能没有鞍点。

测试用例：

输入

3 4

1 3 5 3

2 3 4 1

3 2 6 7

输出

Point:a[1][2]==4

```

#include<stdio.h>
int main()
{
    int n,m,i,j,k,a[100][100],x,y,cont=0;
    scanf("%d%d",&n,&m);
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
}

```

```

for(i=0;i<n;i++)
{
    for(x=0,j=0;j<m;j++)
    {
        if(a[i][j]>a[i][x])
        {
            x=j;
        }
    }
    for(k=0;k<n;k++)
    {
        if(a[k][x]<a[i][x])
        {
            goto mark;
        }
    }
    printf("Point:a[%d][%d]==%d\n",i,x,a[i][x]);
    cont++;
    mark::;
}
if(cont==0)
{
    printf("No Point\n");
}
return 0;
}

```