# **C语言初级阶段结课考试**

**1、以下4组用户标识符中，全部符合标识符命名规则的一组是（ D）。**

1. **-sum,FOR,Case**
2. **3D,Size,abc#**
3. **IF,F4\_66,int**
4. **a\_b,VOID,fun**

**2、以下代码的运行结果是（ B）**

**float a = 123.456789;**

**printf("%2.3f", a);**

1. **123.456 B) 123.457 C) 123.456789 D) 23.456**

**3、设int a=9，则执行完语句a+=a-=a/2后，a的值为（ D）。**

1. **14 B) 9 C) 10 D)0**

**4、以下代码的运行结果是：（ B）**

**int x=5,y=10;**

**x=x+y;**

**y=x-y;**

**x=x-y;**

**printf(“%d”,x>y?x:y);**

1. **5 B)10 C) 15 D) 20**

**5、执行以下语句后，x,y,z的值分别为（ C）。**

**int x,y,z;**

**x=y=z=0;**

**++x||y++&&++z;**

1. **1,1,0 B) 1,1,1 C) 1,0,0 D) 1,0,1**

**6、以下程序的输出结果是（ A）。**

**main()**

**{**

**int n=7;**

**while(--n)printf(“%d\t”,n--);**

**}**

1. **6 4 2 B) 7 5 3 1 C) 6 4 2 0 D) 5 3 1**

**7、执行以下代码后，输出结果为（ B）**

**int arr[] = { 1, 2, 3, 4, 5, 6 };**

**int \*p;**

**p = arr;**

**p += 1;**

**\*(p + 3) -= -5;**

**printf("%d",p[1]);**

1. **2 B) 3 C) 6 D) 10**

**8、执行以下代码后，其输出结果为（ A）。**

**int a[3][2] = { (1, 2), (3, 4), (5, 6) };**

**int \*p = a[1];**

**printf("%d\n", p[1]);**

1. **0 B) 3 C)4 D)5**

**9、以下程序段的输出结果为（B ）。**

**int n = 0, a[3][3] = { 9, 8, 7, 6, 5, 4, 3, 2, 1 };**

**for (;;)**

**{**

**if (n >= 3)break;**

**printf("%d %d,", (\*(a+2 - n))[n],(a+n)[0][2 - n]);**

**n++;**

**}**

1. **7 7,5 5,3 3 B)3 7,5 5,7 3 C)7 5,3 3,5 7 D)3 3,5 5,7 7**

**10、下面三个printf语句分别输出 6 、 4 、 1 。**

**int n[3][4] = { 1, 2, 3, 4, 8, 7, 6, 5, 9, 10, 11, 12 };**

**int \*p[3] = { n[2], n[0], n[1] };**

**printf("%d\n", \*(\*(p + 2) + 2));**

**printf("%d\n", p[1][3]);**

**printf("%d\n", \*(p)[1]);**

**11、函数的递归调用是通过一个函数直接或间接调用此函数本身来实现的，那么在递归调用中必须要有处理无限递归（死递归）的方法，这个说法是否正确？（A ）**

1. **是**
2. **否**
3. **不一定**
4. **不知道**

**12、以下关于int (\*ptr)();的说法中正确的是（C ）。**

1. **ptr是一个指向一维数组的指针变量。**
2. **ptr是一个返回值为int\* （整型指针）类型的指针函数。**
3. **ptr是一个指向函数的指针，用于指向返回值为int类型且没有形参的函数。**
4. **ptr是一个指向()括号的指针，用于指向括号中的一个整型变量。**

**13、以下关于栈和队列的说法正确的是（ C）。**

1. **栈是一种遵循先进先出、后进后出规则的线性数据结构。**
2. **队列是一种遵循先进后出、后进先出规则的线性数据结构。**
3. **栈和队列是功能受限的线性数据结构。**
4. **栈只能用链表实现，而队列只能用顺序表实现。**

**14、链表是一种非常实用的线性数据结构，以下能够使用结构体指针\*p引用其下一个节点数据域data的语句是（C ）。**

1. **p->next.data;**
2. **p.next->data;**
3. **p=p->next;p->data;**
4. **p=\*(p)->next;\*(p).data;**

**15、如有以下定义，则printf语句输出结果是（ A）。**

**#define fun(x,y) x\*x+y\*y**

**printf("%d\n",fun(1+2,fun(3,4)));**

1. **174 B)643 C)639 D)164**

**16、建立函数的目的不包括（ C）。**

1. **提高程序的可读性**
2. **提高代码的复用性**
3. **减少程序的篇幅**
4. **提高程序的执行效率**

**17、C语言规定，以简单变量做函数实参时，它和其对应形参之间的数据传递方式为（ C）。**

1. **地址传递 B)单向值传递 C)由实参传递给形参，再由形参传回给实参 D)由用户指定传递方式**

**18、实现一个对单链表进行冒泡排序的函数，以链表的头节点指针作为函数参数传递（头节点不存储数据）。**

//冒泡排序

for (int i = 0; i < list->length; i++)

{

for (Node\* temp2 = list->head; temp2; temp2 = temp2->next)

{

if (temp2->next != NULL)

if (temp2->data > temp2->next->data)

{

int t = temp2->data;

temp2->data = temp2->next->data;

temp2->next->data = t;

}

}

}

**19、实现一个算法，用于求一个一元二次方程（a\*x\*x+b\*x+c=0）的实数解(有两个解输出解x1和x2，只有一个解的输出x的值，否则输出无解)。**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

#include <math.h>

int main()

{

float a, b, c;

scanf("%f%f%f", &a, &b, &c);

if (b\*b - 4 \* a\*c < 0)

{

printf("此方程无解！\n");

}

else if (b\*b - 4 \* a\*c > 0)

{

if (a == 0)

{

printf("此方程只有一解, X1 = X2 = %.2f", -c / b);

}

else

{

float x1 = (-b + sqrt(b\*b - 4 \* a\*c)) / (2 \* a);

float x2 = (-b - sqrt(b\*b - 4 \* a\*c)) / (2 \* a);

printf("此方程有两个解，X1 = %.2f,X2 = %.2f\n", x1, x2);

}

}

else

{

float x1 = -b / (2 \* a);

printf("此方程只有一解, X1 = X2 = %.2f", x1);

}

return 0;

}

**20、求出100以内的素数**

**素数是指大于1的自然数中，除了1和其本身以外，不能被其他自然数整除（除0以外）的称为素数（质数）**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

#include <math.h>

bool prime(int n)

{

int i = 0;

for (i = 2; i < sqrt(n); i++)

{

if (n % i == 0)

return false;

}

return true;

}

int main()

{

int i = 0;

for (i = 2; i <= 100; i++)

{

if (prime(i))

{

printf("%d ", i);

}

}

return 0;

}