

《C 程序设计基础》

2021-2022 学年第一学期期中考试试卷（一）

一、判断题

1. 执行以下程序段，输入 20，输出 20.00。

```
double x;  
scanf("%f",&x);  
printf("%.2f",x);
```

2. 下面这段代码，用%.2f 格式打印输出 int 型变量，不会进行类型转换，打印输出的结果不会是 2019.00。

```
#include <stdio.h>  
int main(){  
    int k=2019;  
    printf("%.2f", k);  
    return 0;  
}
```

3. 执行以下程序段后，x 的值为 0，y 的值为 1。

```
int x,y;  
if(x = 0) y = 0;  
else y = 1;
```

4. 为了检查以下 if-else 语句的两个分支是否正确，至少需要设计 3 组测试用例，即 x 的取值至少有 3 组（小于 15 的数、15 和大于 15 的数）。

```
if(x <= 15){  
    y = 4 * x / 3;  
} else{  
    y = 2.5 * x - 10.5;  
}
```

5. 如果变量已经正确定义，则执行以下程序段后的 x 值不变。

```
x = 4;  
if(x < 0){  
    y = -1;  
} else if(x = 0){  
    y = 0;  
} else{  
    y = 1;  
}
```

6. 设变量已正确定义，以下是合法的 switch 语句。

```
switch(choice){  
    case choice==1:price = 3.0;break;  
    case choice==2:price = 22.5;break;  
    case choice==3:price = 4.0;break;  
    case choice==4:price = 3.5;break;  
    default:price =0.0;break;  
}
```

7. continue 智能用于循环中。

8. 若变量已正确定义，执行以下 while 语句将陷入死循环。

```
i=1;
while(i<=10);
i++;
```

9. 以下两个程序段不等价，执行程序段 B 将陷入死循环

/*程序段 A*/

```
s = 0;
for (i = 1; i <= 10; i++)
{
    if (i % 2 == 0)
    {
        continue;
    }
    s = s + i;
}
```

/*程序段 B*/

```
s=0;
i=1;
while (i<=10)
{
    if(i%2==0){
        continue;
    }
    s=s+i;
    i++;
}
```

10. 执行以下程序段，将出现死循环。

```
for(i=1;i<=20;i++){
    for(i=1;i<=10;i++){
        printf("%4d",i);
    }
    printf("\n");
}
```

11. 函数形参的存储单元是动态分配的。
12. 函数的实参和形参都可以是变量、常量和表达式。
13. 按照 C 语言的规定，实参和形参的命名不得重复。
14. 全局变量与局部变量的作用范围相同，不允许它们同名。
15. 当多个函数发生多层嵌套调用时，最后被调用的函数却最先结束。
16. 一个函数定义中可以完整地包含另一个函数的定义。
17. 一个被调函数的函数体中，又可以出现函数调用语句，这种调用现象称为函数的嵌套调用。
18. 在定义数组之后，根据数组中元素的类型及个数，在内存中分配一段连续存储单元用于存放数组中的各个元素。
19. 数组定义后，数组名的值可以是一个地址，可以被修改。
20. 数组初始化时，初始值个数小于数组元素的个数，C 语言自动将剩余的元素初始化为初始化列表中的最后一个初始值。

二、选择题

1. 下面四个选项中，均不是合法的用户标识符的选项是（ ）
 - A. A 、 P_0 、 do
 - B. float 、 1a0 、 _A
 - C. b-a 、 goto 、 int
 - D. _123 、 temp 、 INT
2. 使用“scanf("X=%f,y = %f", &x , &y)”，要使 X, y 均为 1.25，正确的输入是()
 - A. 1.25,1.25
 - B. 1.25 1.25
 - C. X=1.25,y=1.25
 - D. X=1.25 y=1.25
3. 若变量 a 是 int 类型，并执行了语句：a='A'+1.6 ;则正确的叙述是
 - A. a 的值是字符 C
 - B. a 的值是浮点型
 - C. a 的值是字符'A'的 ASCII 值加上 1。
 - D. 不允许字符型和浮点型相加
4. 设 int x=1,y=1;表达式(!x||y--)的值是（ ）
 - A. 0
 - B. 2
 - C. 1
 - D. -1
5. 要求以下程序的功能是计算：s=1+1/2+1/3+...+1/10，但运行的结果错误，导致错误结果的程序行是（ ）

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

```
int main(void) {  
    int n;float s;  
    s = 1.0;  
    for (n = 10; n > 1; n--)s = s + 1 /n;  
    printf("%6.4f\n", s);  
    return 0;  
}
```

- A. int n; float s;
 - B. for (n = 10; n > 1; n--)
 - C. s = s + 1 /n;
 - D. s = 1.0;
6. 已知 int x=10,y=20,z=30;以下语句： if(x<y) z=x; else x=y; y=z;执行后 x,y,z 的值是（ ）
 - A. x=10,y=10,z=10
 - B. x=20,y=30,z=30
 - C. x=20,y=30,z=10
 - D. x=20,y=20,z=20
 7. 如果变量 x,y 已经正确定义，下列哪个选项的语句不能正确将的 x,y 的值进行交换？
 - A. x=x+y, y=x-y, x=x-y;
 - B. t=x,x=y; y=t;
 - C. t=y,y=x, x=t;
 - D. x=t, t=y, y=x;

8. 不正确的赋值或赋初值的方式是____

- A. `char str[]="string";`
- B. `char str[7]={ 's', 't', 'r', 'i', 'n', 'g' };`
- C. `char str[10]; str="string"`
- D. `char str[7]={ 's', 't', 'r', 'i', 'n', 'g', '\0' };`

9. 设置变量已正确定义，以下（ ）与其它 switch 语句 不等价。

- A.

```
switch(choice){
    case 1:price = 3.0;break;
    case 2:price = 2.5;break;
    case 3:price = 4.0;break;
    case 4:price = 3.5;break;
    default:price =0.0;break;
}
```
- B.

```
switch(choice){
    default:price =0.0;break;
    case 4:price = 3.5;break;
    case 3:price = 4.0;break;
    case 2:price = 2.5;break;
    case 1:price = 3.0;break;
}
```
- C.

```
switch(choice){
    case 1:price = 3.0;break;
    case 2:price = 2.5;break;
    case 3:price = 4.0;break;
    case 4:price = 3.5;break;
}
```
- D.

```
price = 0.0;
switch(choice){
    case 1:price = 3.0;break;
    case 2:price = 2.5;break;
    case 3:price = 4.0;break;
    case 4:price = 3.5;break;
}
```

10. 下列程序段的功能是：计算数组 x 中相邻两个元素的和，一次存放到 a 数组中，然后输出 a 数组。程序段中待填空的（1）和（2）的正确选项是（ ）。

```
int i;
int a[9], x[10];
for (i = 0; i < 10; i++)
{
    scanf("%d", &x[i]);
}
for ( (1) ; i < 10; i++)
{
    a[i - 1] = x[i] + (2) ;
}
for (i = 0; i < 9; i++)
{
```

```
        printf("%d ", a[i]);
    }
    printf("\n");
```

- A. (1)i=1(2)x[i+1]
- B. (1)i=0(2)x[i-1]
- C. (1)i=1(2)x[i-1]
- D. (1)i=0(2)x[i+1]

三、代码分析

1. 程序 A 和程序 B 的功能是：

<pre>/*程序段 A*/ #define MAXN 10; int i, high, low, temp; int num[MAXN]; scanf("%d", &n); for (i = 0; i < n; i++) { scanf("%d", &num[i]); } high = 0; low = n - 1; while (high < low) { temp = num[high]; num[high] = num[low]; num[low] = temp; high++, low--; } for (i = 0; i < n; i++) { printf("%d ", num[i]); } printf("\n");</pre>	<pre>/*程序段 B*/ #define MAXN 10; int i, n, temp; int num[MAXN]; scanf("%d", &n); for (i = 0; i < n; i++) { scanf("%d", &num[i]); } for (i = 0; i < n / 2; i++) { num[i] = num[i] + num[n - i - 1], num[n - 1 - i] = num[i] - num[n - 1 - i], num[i] = num[i] - num[n - 1 - i]; } for (i = 0; i < n; i++) { printf("%d ", num[i]); } printf("\n");</pre>
--	--

三、编程题

1. 冒泡法排序：本题要求将给定的 n 个整数从小到大排序后输出。输出时相邻数字中间用一个空格分开，行末不得有多余空格。

2. 弹球距离：设有一个球从高度为 h 米的地方落下，碰到地面后又弹到高度为原来 p 倍的位置，然后又落下，再弹起…。请编写函数求初始高度为 h 的球下落后到基本停下来（高度小于给定阈值 TOL ）时在空中所经过的路程总和。

函数接口定义： <code>double dist(double h,double p);</code> TOL: <code>#define TOL 1E-3;</code>
其中 h 是球的初始高度， p 是球弹起高度与弹起前落下高度的比值；函数要返回球下落后到基本停下来时在空中所经过的路程总和。 注意：当弹起的高度小于裁判程序定义的常数 TOL 时，弹起的距离不计算在内。

3. 已知输入正整数 $m,n(m<n)$ ，请输出 m, n 之最大素数和最小素数及所有素数的和。1 不是素数，如果没有素数，则输出“No primes”。

输入格式	在一行输入两个正整数 m,n
输出格式	在一行中输出结果，格式如“ <code>Max=###,Min=###,Sum=###.</code> ”，“No primes.”或 “Error input.”(如果 $m,n<0$).

2021-2022 学年第一学期期中考试试卷（一）参考答案

一、判断题

1. 【正解】错误

【解析】因为 x 为双精度浮点数，但是输入函数中是 `%f`，为单精度 `float` 的输入标志。在输入时会发生转换错误，20 并不会赋值到 x 。应将 `%f` 改为 `%lf`。

【考点延伸】《考试宝典》1.3 格式化输入与输出

2. 【正解】正确

【解析】原因与第一题类似， k 为整型，而输出为浮点型，无法由低精度向高精度转换。

【考点延伸】《考试宝典》1.3 格式化输入与输出

3. 【正解】正确

【解析】`if(x=0)`，该语句内是一个赋值表达式，此时 $x=0$ ，进入 `else`， $y=1$ 。

【考点延伸】《考试宝典》3.2——`if` 语句

4. 【正解】正确

【解析】 $y=4*x/3$ 和 $y=2.5 * x - 10.5$ 在各自定义域内并无相同的函数值，因此只需测试三组。

【考点延伸】《考试宝典》3.2——`if` 语句

5. 【正解】错误

【解析】在第二个判断语句中， $x=0$ 是一个赋值表达式，因此 x 的值变为 0。

【考点延伸】《考试宝典》3.2——`if` 语句

6. 【正解】错误

【解析】`case` 后面直接跟具体的数值，例如 `case 1:~`。

【考点延伸】《考试宝典》3.3——`switch` 语句

7. 【正解】正确

【解析】`continue` 用于结束当次循环。

【考点延伸】《考试宝典》循环控制关键字 `continue` 关键字

8. 【正解】正确

【解析】会一直在 `while` 循环内，无法进入到 `i++`， i 值无法改变。

【考点延伸】《考试宝典》4.1——`while` 语句

9. 【正解】正确

【解析】B 中第一次循环， $i=1$ 时，不会执行 `i++`， i 值永远不变，进入死循环。

【考点延伸】《考试宝典》专题三、四——循环结构与 `if` 语句。

10. 【正解】正确

【解析】进入第二层循环后， i 的值最后都会变为 10。

【考点延伸】《考试宝典》专题四——循环结构

11. 【正解】正确

【解析】函数形参的存储单元是动态分配的

【考点延伸】《考试宝典》专题八——函数

12. 【正解】错误

【解析】函数的形参可以设置默认值，但不能是常量或者表达式。

【考点延伸】《考试宝典》专题八——函数

13. 【正解】错误

【解析】可以重复。

【考点延伸】《考试宝典》专题八——函数

14. 【正解】错误

【解析】不相同，全局变量的作用于全局，即整个源程序；局部变量作用于局部，如函数体内，代码块内；对于函数体外局部变量没有作用。可以同名。

【考点延伸】《考试宝典》专题七——变量的作用域

15. 【正解】正确

【解析】逐层调用，最后被调用的函数最先执行，也最先结束。

【考点延伸】《考试宝典》专题八——函数

16. 【正解】错误

【解析】函数体内可以有并一个函数的声明，但是不能完整包含另一个函数的定义。

【考点延伸】《考试宝典》专题八——函数

17. 【正解】正确

【解析】函数体内出现调用 A 函数，A 函数中又调用了 B 函数，。这种现象就叫函数的嵌套调用。

【考点延伸】《考试宝典》专题八——函数

18. 【正解】正确

【解析】数组的定义。分配一段连续数据单元存放元素。

【考点延伸】《考试宝典》专题五——数组

19. 【正解】错误

【解析】数组定义后，数组名不可更改，数组大小不可更改，数据类型不可更改。

【考点延伸】《考试宝典》专题五——数组

20. 【正解】错误。

【解析】对于全局数组，剩余的值为 0；而局部数组，剩余的元素值随机。

【考点延伸】《考试宝典》专题五——数组

二、选择题

1. 【正解】C

【解析】合法的标识符只能由 26 个字母、_、数字组成，不能以数字开头，区分大小写。不能是关键字、保留字。

【考点延伸】《考试宝典》1.1——标识符

2. 【正解】C

【解析】严格遵守 $X=\%f, y=\%f$ 的格式，中间有逗号，选 C

【考点延伸】《考试宝典》1.3 格式化输入输出

3. 【正解】C

【解析】在运算过程中，都会转化为 int 型。

【考点延伸】《考试宝典》专题二——数据类型的转换

4. 【正解】C

【解析】 $!x=0$ ；y--先用 y 的值 1 后，再执行 y-1。经过或运算，结果为 1。

【考点延伸】《考试宝典》2.2 运算符和表达式

5. 【正解】C

【解析】应将 int 型转为 float 型进行计算。 $s = s + 1 / (\text{float}) n$ ；

【考点延伸】《考试宝典》专题二——数据类型的转换

6. 【正解】A

【解析】 $x < y$ 成立， $z=10$ ；不进入 else； $y=z=10$ 。

【考点延伸】《考试宝典》3.2——if 语句

7. 【正解】D

【解析】D 中交换的是 t 和 y 的值。

【考点延伸】《考试宝典》2.2 运算符的运用

8. 【正解】C

【解析】不能通过 `str="string"` 方式赋值

【考点延伸】《考试宝典》5.4 字符数组

9. 【正解】C

【解析】当 choice 不为 1、2、3、4 时，其它三个 price 都为 0.0。而 C 没有默认值。

【考点延伸】《考试宝典》3.3——switch 语句

10. 【正解】C

【解析】(1) 因为循环体里面是从 i-1 开始，所以此处为 i=1；(2) a[i-1] 等于 x[i]加上它的前一个元素即 x[i-1]。

【考点延伸】《考试宝典》4.3 循环 for

三、

1. 【正解】以下程序段 A 和 B 的功能都是：将给定的 n 个整数存入数组中，将数组中的这 n 个数逆序存放，再按顺序输出数组中的这 n 个数。

【考点延伸】《考试宝典》宏定义、数组的综合算法。

四、编程题

1. 【解析】

```
#include <iostream>
#include <vector>

using namespace std;

int main() {
    vector<int> nums;
    int n, temp;
    printf("Please enter the array length: ");
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> temp;
        nums.push_back(temp);
    }
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - 1 - i; j++) {
            if (nums[j] > nums[j + 1]) {
                temp = nums[j];
                nums[j] = nums[j + 1];
                nums[j + 1] = temp;
            }
        }
    }
    for (int i = 0; i < n - 1; i++) {
        cout << nums[i] << " ";
    }
    cout << nums[n - 1] << endl;
    return 0;
}
```

【考点延伸】重要算法——冒泡排序

2. 【解析】

```
double dist(double h, double p)
{
    double result, hight;
```

```

    hight = h;
    result = h * (-1);
    if (h < TOL)
    {
        result = h;
    }
    else
    {
        while (hight >= TOL)
        {
            result += 2 * hight;
            hight *= p;
        }
    }
    return result;
}

```

【考点延伸】函数、if 语句和循环的综合应用

3. 【解析】

```

#include <iostream>
#include "math.h"

using namespace std;

bool isPrime(int num) {
    if (num < 2)
        return false;
    int flag = -1;
    for (int i = 2; i < num; i++)
        if (num % i == 0)
            flag = i;
    if (flag > 0)
        return false;
    return true;
}

int main() {
    int m, n, sum = 0, count = 0;
    cin >> m >> n;
    if (m < 0 || n < 0)
        printf("Error input.");
    int max = n, min = m;
    if (m < 2) min = 2;
    for (int i = m; i <= n; i++) {
        if (isPrime(i)) {
            sum += i;
            count++;
        }
    }
}

```

```
        if (i > max)
            max = i;
        if (i < min)
            min = i;
    }
}
if (count == 0) {
    printf("No primes.");
} else {
    printf("Max=%d,Min=%d,Sum=%d", max, min, sum);
}
return 0;
}
```

【考点延伸】素数判断算法。

发现错误怎么办

反馈有奖



扫码或联系QQ: 1152296818

本资料编者都是学长学姐，虽然仔细核对了很多遍，但可能会有一些疏漏，诚恳希望学弟学妹们积极反馈错误，我们会及时更正在二维码里哦 (づ￣3￣)づ