

## 2017 级 C 语言模拟测试试题参考解答

一、判断下列语句或程序段的对错。（“×”表示错，“√”表示对）（10 分）

//评分：每小题 1 分

- (1) float x=0.23e+09L (×)
- (2) unsinged int a=b=c='0'; (×)
- (3) printf(“%f\n”,(40,50,60)); (×)
- (4) int data[4]={0,1,2,3,4}; (×)
- (5) char \*p=“hust66”;  
gets(p); (×)
- (6) #define N = 50;  
int a[N]; (×)
- (7) char str[]=”hust66”;  
printf(“%c”,\*(str+2)); (√)
- (8) int x = (018 == 'A') ; (×)
- (9) float a=1, \*b=&a, \*\*c=b; (×)
- (10) float \*p,a[2][3];  
p=\*a + 2; (√)

二、计算下列表达式的值（10 分） //评分：每小题 2 分

设 unsigned int x=2.5,y=4.7,a=7,b=17,c=4, d=3;

float f;

- (1) f=b/c (4.0)
- (2) !(a+b)+c-1&&b+c/2 (1)
- (3) (a^b) + (c>>1|d) (25)
- (4) a += b %= a+b (24)
- (5) a=2,b = a \* ++b (36)

三、改错，根据题意改正下列程序的错误和漏掉的部分（10 分）

(1) 求两个浮点数的平方和及平方差.(5 分)

#include <stdio.h>

float calculate(float,float,float\*) 1 分

void main()

{

float a,b;

float add\_result,sub\_result;

scanf(“%f,%f”,a,b);

scanf(“%f,%f",&a,&b); 1 分

add\_result=calculate(a,b,&sub\_result);

printf(“a\*a+b\*b=%d,a\*a-b\*b=%d\n”,add\_result,sub\_result); %d 改成%f 1 分

```
}
```

calculate (float a,float b,float \*sub)      float calculate (float a,float b,float \*sub)      0.5 分

```
{
```

float \*temp;      float temp;      0.5 分

```
*sub=a*a - b*b;
```

```
*temp=a*a + b*b;      temp=a*a+b*b;      0.5 分
```

```
return *temp;      return temp;      0.5 分
```

```
}
```

(2)统计 N 个字符串中大写字母和数字字符的个数(5 分)

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

```
int Count (char *str,int *result);      0.5 分
```

```
#define N      5
```

```
main()
```

```
{
```

```
char string[N][80];
```

```
char i;
```

```
int Capital_Count,Num_Count;      int Capital_Count=0,Num_Count=0;      0.5 分
```

```
for(i=0;i<=N;i++)      for(i=0;i<N;i++)      0.5 分
```

```
scanf("%s",&string[i]);      scanf("%s",string[i]);      1 分
```

```
for(i=0;i<N;i++)
```

```
Capital_Count+=Count(string[i],&Num_Count);
```

```
printf("Capital count :=%d,numbercount=%d\n"      \
```

```
,Capital_Count,Num_Count);
```

```
}
```

Count(char \*str,int \*result)      int Count(char \*str,int \*result)      0.5 分

```
{
```

```
int temp,i;      int temp=0,i;      0.5 分
```

```
for(i=0;i<80;i++)      for(i=0;str[i]!='\0';i++)      0.5 分
```

```
{
```

```
if(str[i]>='A'&& str[i]<='Z')
```

```
temp++;
```

```
if(str[i]>'0' || str[i]<'9')      if(str[i]>='0'&&str[i]<='9')      0.5 分
```

```
*result++;      (*result)++;      0.5 分
```

```
}
```

```

        return temp;
    }

```

#### 四、程序填空（10 分）（每空 1 分）

（1）按如图所示的格式打印杨辉三角形

```

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

```

.....

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

```
void main()
```

```
{
```

```
    int a,b,m[10][10];
```

```
    for(a=0;a<10;a++)
```

```
    {
```

```
        m[a][0]=1;
```

```
        m[a][a]=1;
```

```
    }
```

```
    for(a=2;a<10;a++)
```

```
        for(b=1;b<a;b++)
```

```
            m[a][b]=m[a-1][b-1]+m[a-1][b];
```

```
    for(a=0;a<10;a++)
```

```
    {
```

```
        for(b=0;b<=a;b++)
```

```
            printf("%5d",m[a][b]);
```

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

```
    }
```

}

(2) 利用公式  $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!}$  计算  $\sin x$  的值, 设

$x=0.4$  时  $\sin x$  的值

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

```
#include <math.h>
```

```
main()
```

```
{
```

```
    float x,sn,un;
```

```
    int i,j;
```

```
    x=0.4;
```

```
    un=_____x_____;
```

```
    sn=_____x 或 un_____;
```

```
    i=1;
```

```
    while(fabs(un)>=1e-5)
```

```
    {
```

```
        un=_____un*(-1)*x*x/(2*i)/(2*i+1);
```

```
        sn=_____sn+un_____;
```

```
        i++;
```

```
    }
```

```
    printf("x=%f,sinx=%f\n",x,sn);
```

```
}
```

五、写程序输出结果 (25 分) //评分: 每小题格式错误扣 1 分, 第(3)小题结果错一个扣 0.5 分, 其他小题错一个扣 1.5 分。

(1)

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

```
void fun(int *,int);
```

```
void main()
```

```
{
```

```
    int a[]={1,2,3,4},i;
```

```
    fun(a+1,2);
```

```
    for(i=1;i<4;i++)
```

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

```
void fun(int *b,int n)
{
    int i;
    for(i=0;i<n;i++)
        b[i]=b[i]*b[i];
    return ;
}
```

答案:

4

9

4

( 2 )

```
#include <stdio.h>
void main()
{
    int i,j,min;
    int row=0,column=0;
    int a[3][3]={ {1,2,3},{2,-3,4},{9,4,7}};

    min=a[0][0];
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
        {
            if(min >a[i][j])
            {
                min=a[i][j];
                row=i+1;
                column=j+1;
            }
        }
    printf("min=%d,row=%d,column=%d\n",min,row,column);
}
```

答案:

min=-3,row=2,column=2

( 3 )

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

```
int d;
```

```
void func();
```

```
void main()
```

```
{
```

```
    func();
```

```
    func();
```

```
    func();
```

```
}
```

```
void func()
```

```
{
```

```
    static int a=2;
```

```
    int b=0;
```

```
    printf("a=%4d\nb=%4d\nd=%4d\n",a++,b++,d++);
```

```
}
```

答案 : (以下数字前都有三个空格)

a= 2

b= 0

d= 0

a= 3

b= 0

d= 1

a= 4

b= 0

d= 2

( 4 )

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

```
struct Key
```

```
{
```

```
    char *keyword;
```

```
    int keyno;
```

```
};
```

```

void main( )
{
    struct Key   kd[3] = { {"are",123}, {"your",456}, {"my", 789} };
    struct Key *p;
    int a;
    char *str;

    p = kd;
    str = p++->keyword;
    printf("str = %s\n",str+1);

    a = ++p->keyno;
    printf("a = %d\n",a);

    p=kd;
    a = p->keyno;
    printf("a = %d\n",a);
}

```

答案:

str = re

a = 457

a = 123

(5)

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

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

```
void main()
```

```

{
    char *name[]={“Java”,“Basical”,“windows”,“TurboC++”};
    int a,b,n=4;
    char *temp;

    for(a=0;a<n-1;a++)
        for(b=a+1;b<n;b++)
        {
            if(strcmp(name[a],name[b])<0)
            {
                temp=name[a];
                name[a]=name[b];

```

```

        name[b]=temp;
    }
}
for(a=1;a<n;a++)
    printf("%s\n",name[a]);
}

```

答案:

TurboC++

Java

Basical

六、编写程序（35 分）(编程过程中，不得使用全局变量)

(1) 编一程序，输入 x 的值，输出 y 的值。（9 分）

$$y = \begin{cases} x & x < 1 \\ 2x-2 & 1 \leq x < 10 \\ 3x-10 & x \geq 10 \end{cases}$$

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

//头文件包含 1 分

```
void main()
```

//函数名和变量定义 1 分

```
{
```

```
    float x,y;
```

```
    printf("Please enter the x(integer):");
```

//输入 1 分

```
    scanf("%f",&x);
```

```
    if(x<1)
```

//算法 5 分

```
        y=x;
```

```
    else if(x>=10)
```

```
        y=3*x-10;
```

```
    else
```

```
        y=2*x-2;
```

```
    printf("y=%f",y);
```

//输出 1 分

```
}
```

(2)编写一个函数求一个区间的所有素数，要求区间在主函数输入，调用所编函数求出的素数在主函数中显示(9 分)

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

//头文件、函数声明、宏定义 1 分



```

#define N 100 //长度可调整
int calc_primer(int *primers,int a,int b);
void main()
{
    int a,b;
    int i;
    int primers[N];
    int primer_num;
    printf("please enter two number:");
    scanf("%d%d",&a,&b); //输入 1 分
    primer_num=calc_primer(primers,a,b); //函数调用 1 分
    for(i=0;i<primer_num;i++) //输出 1 分
    {
        printf("%5d",*(primers+i));
        if((i+1)%5==0)
            puts("");
    }
}

int calc_primer(int *primers,int a,int b) //函数原型 1 分
{
    int cnt=0;
    int i;
    int num;

    for(num=a;num<=b;num++) //算法 3 分
    {
        for(i=2;i<num;i++)
        {
            if(num%i==0)
                break;
        }
        if(i==num)
            primers[cnt++]=num;
    }
    return cnt; //返回 1 分
}

```

- 3) 编写一个函数，void fun(char \*str, char \*substr1, char \*substr2, ...)实现将字符串 str 中的子串 substr1 替换为子串 substr2，并同时统计替换后 str 字符串长度，要求：接口定义中已有部分不许修改，但允许在...处自由添加参数。替换举例：如将字符串“abcdfrbcxybc”中的子串 1“bc”替换为子串 2“ghi”后为“aghiedfrghixyghi”，主函数 main 完成字符串和子串的输入，调用函数 fun 得到替换后的字符串，最后在主函数中输出新串及由 fun 函数计算出来的长度 16。如果子串 substr1 不在 str 中，则 fun 函数只统计字符串长度。（9 分）

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

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

```
void fun(char *str, char *substr1, char *substr2 , int *plen);//第四个参数可以 int& reflen
```

```
//对应 main 中调用格式: fun( string, str1, str2, length );
```

```
//fun 函数体中最后一条执行语句改为: reflen=strlen(str);
```

```
-----头文件和原型申明共 2 分
```

```
void main( void )-----共 3 分
```

```
{
char string[100] = "abcdfrbcxybc";
char str1[ ] =    "bc";
char str2[ ] =    "ghi";
    int  length;
    printf( "before:%s and total size:%d\n", string ,strlen(string));
    fun( string, str1, str2, &length );
    printf( "after:%s and total size: %d\n", string, length );
}
```

```
void fun(char *str, char *substr1, char *substr2 , int *plen)---接口定义 1 分
```

```
{
char *pfound=NULL;
int sub1len,sub2len;
```

```
sub1len =strlen(substr1);
```

```
sub2len =strlen(substr2);
```

```
pfound = strstr(str,substr1);//调用 strstr 库函数查找主串 str 中是否存在 substr1 子串，返回 NULL 指针（表明未找到）或指向找到的子串存储空间入口的指针
```

```
while (pfound != NULL)-----查找&替换 2 分
```

```
{
    char temp[100];

    strcpy(temp,pfound+sub1len);//将对应子串后面的字符内容搬移到 temp
    strcpy(pfound,substr2);//通过字符串拷贝，用子串 2 替换掉子串 1 内容
    strcat(str,temp);//strcpy(pfound+sub2len,temp);//strcat(pfound,temp);
    //上面三种写法实现将 temp 串追加到替换后的子串 2 串结尾符\0 后面
```

```
    pfound = strstr(pfound +sub2len, substr1);
}
```

```
*plen =strlen(str);-----1 分
```

```
return ;
```

```
}
```

- 4) 设有 N 个参加复试的考生，考生信息包括编号、姓名、性别和 4 门功课的成绩，main 函数负责输入各单科录取线及考生信息，并统计每个学生的总分。编写函数实现：(1) sort 函数实现按总分进行排序（降序）；(2) search 函数找出至少有一门课低于单科录取线的所有考生信息；(3) print 函数实现按总分从高到低打印排名前 20%的单科达到要求的考生信息；要求：排序、查找和打印输出分别用函数实现。（8 分）

```
#include <stdio.h>
#include <stdlib.h>
#define N    3
typedef struct
{
    char s_id[20];
    char s_name[20];
    char sex;
    float score[4];
    int status;//记录状态。0：所有单科达标 1：单科未达标
    float total;
} STU;

//-----以上 1 分
void bubblesort(STU *data,int n);
int search(STU *data, int students, float * danke, int courses, STU *select);
void print(STU *data, int students, double percent);//调用前先执行降序操作
//-----3 个原型申明 1 分

int main( )                // -----2 分
{
    STU stu[N];

    int i,j;
    float danke[4];
    char stemp[20];
    STU sel[N];
    int found;

    printf("录入并统计每个学生的总分并记录:\n");
    for(i=0;i<N;i++)
    {
        printf("No%d:\n",i+1);

        printf("id(a string):");
        gets(stu[i].s_id);

        printf("Name(a string):");
```

```

        gets(stu[i].s_name);

        printf("Sex(a char):");
        gets(stemp);
        stu[i].sex = stemp[0];

        stu[i].total=0.0;

        for(j=0;j<4;j++)
        {
            printf("Score %d (a float data):" ,j+1 );
            gets(stemp);
            stu[i].score[j] = atof(stemp);
            stu[i].total += stu[i].score[j];
        }
    }

    for(i=0;i<4;i++)
    {
        printf("录入单科%d 录取线:",i+1);
        gets(stemp);
        danke[i] = atof(stemp);
    }

    found = search(stu,N,danke,4,sel);
    if(found>=1)
    {
        printf("共有%d 个学生未达到单科线要求!\n",found);
        for(i=0;i<found;i++)
        {
            printf("%10s %20s %6.2f %8.2f%8.2f%8.2f\n", \
                sel[i].s_id, \
                sel[i].s_name,sel[i].total, \
                sel[i].score[0],sel[i].score[1], \
                sel[i].score[2],sel[i].score[3]);
        }
    }
    else
        printf("所有学生通过复试线\n");

    bubblesort(stu,N);//降序排列总分

    puts("打印总分排名前 20%的单科达标学生信息:");

```

```

print(stu,N,0.2);
return 0;
}

```

```

void bubblesort(STU *data,int n)           // -----2 分

```

```

{
    int i,j;

    for(i=1;i<=n-1;i++)
        for(j=1;j<=n-i;j++)
            if(data[j-1].total < data[j].total)
            {
                STU temp;
                temp = data[j];
                data[j] = data[j-1];
                data[j-1]= temp;
            }
}

```

```

int search(STU data[ ],int students,float danke[ ],int courses,STU *select)

```

```

//—1 分

```

```

{
    int i,j,k = 0;

    for(i=0;i<students;i++)
    {
        for(j=0;j<courses;j++)
        {
            if(data[i].score[j]<danke[j])
            {
                data[i].status=1;
                select[k++]=data[i];//可以不用参数 select，只需记录状态，在主调函数中
                遍历学生数组 stu 时判断状态，找出所有 status 为 1 的学生，但效率低
                break;
            }
            else
                data[i].status=0;
        }
    }
    return k;
}

```

```

void print(STU *stu,int students,double percent)//调用前先执行降序操作----1 分

```

```

{
int i;
int prints;
double value;
value = students * percent ;
if((int)value == value)
    prints =(int)(value);
else
    prints =(int)(value)+1;

for(i=0;i<prints;i++)
{
    if(stu[i].status==0)
        printf("%10s %20s %8.2f %8.2f %8.2f %8.2f %8.2f\n",\
            stu[i].s_id, \
            stu[i].s_name, \
            stu[i].total,\
            stu[i].score[0],\
            stu[i].score[1],\
            stu[i].score[2],\
            stu[i].score[3]);
}
}

```