

# 2020 级电信学院《计算机与程序设计基础(C)》考试试题 (A)

(考试时间: 2021.01.12 考试时长: 150 分钟)

题目编号	1(15')	2(15')	3(15')	4(15')	5(35')	总分(100')
得分						

## 一、判断下列语句或程序段的对错。(“×”表示错,“√”表示对)(15 分)

- (1) `int x = y = 079;` (×)
- (2) `char a = 'a', c;`  
`c = a++;` (√)
- (3) `register int n; scanf("%d", &n);` (×)
- (4) `int array[5] = {1, 2, 3, 4, 5, 6};` (×)
- (5) `int a[10], *p;`  
`p = a+2;` (√)
- (6) `char *pch`  
`pch = "ABC";` (√)
- (7) `int (*Parr)[8], array[6][8];`  
`Parr = array[0];` (×)
- (8) `int a = 5, b = 7, c;`  
 执行 `c = a+++ (++b)` 语句后, a、b、c 的值分别为 6, 8, 12。 (×)
- (9) `char str[] = {"World"};`  
`printf("%c", *(str+2));` (√)
- (10) `int **pp, *array[6];`  
`pp = &array[0];` (√)
- (11) `int *p1, *p2, m=5, n;`  
`p1 = &m; *p2 = n;` (×)
- (12) `float a=1, *b=&a, **c=&b;` (√)
- (13) `while(1)`  
`if(getchar() == '\n') break;` (√)
- (14) `int x = '\xa8';` (√)
- (15) `int sum(int x, int y)`  
`{int result = x + y; }` (×)

## 二、单选题 (15 分)

- (1) 设整型变量 a、b、c 的值均为 1, 表达式语句 `“++a && ++b || ++c”` 执行后 c 的值是 ( C )  
 A) 3                      B) 2                      C) 1                      D) 0
- (2) 字符串 `“w/x42\\n\\nx\103y”` 在内存中的字节数是 ( B )  
 A) 8                      B) 11                      C) 10                      D) 16

- (3) 能正确表示“当 x 的取值在 [1, 100] 或 [200, 300] 范围内为真，否则为假”的表达式是( C )。
- A) (x>=1) && (x<=100) && (x>=200) && (x<=300)  
 B) (x>=1) || (x<=100) || (x>=200) || (x<=300)  
 C) (x>=1) && (x<=100) || (x>=200) && (x<=300)  
 D) (x>=1) || (x<=100) && (x>=200) || (x<=300)
- (4) 关于字符串比较大小，下面描述正确的是( D )。
- A)字符串所包含的字符个数相同时，才能比较字符串  
 B)字符个数多的字符串比字符个数少的字符串大  
 C)字符串"stop"与"stop"相等  
 D)字符串"That"小于字符串"The"
- (5) 对于“int \*p[5];”的描述，( B )是正确的。
- A)p 是一个指向某数组中第 5 个元素的指针，该元素是 int 类型变量  
 B)p 是一个具有 5 个元素的指针数组，每个元素是一个指向 int 目标的指针  
 C)p 是一个指向数组的指针，所指向的数组有 5 个 int 类型的元素  
 D)p[5]是一个指向 int 目标的指针
- (6) 在针对函数返回值类型和 return 表达式中的表达式结果类型描述中，( D )是错误的。
- A)函数返回值类型是在定义函数时确定的，函数调用时是不能改变的；  
 B)函数返回值的类型就是定义函数时函数名左边的数据类型；  
 C)return 表达式中表达式结果类型与函数返回值类型不同时，return 表达式类型将转换成函数返回值类型；  
 D)当函数不需要返回值时，可以不定义函数返回值类型
- (7) 设有如下定义语句，正确的表达式是( A )。

```
struct node
{
    int len;
    char *pk;
}x = {2, "right"}, *p = &x;
```

- A) p->pk                      B) \*p.pk                      C) (\*p)->pk                      D) \*x.p
- (8) 若有定义语句 int i,x[3][4];则不能将 x[1][1]的值赋给变量 i 的语句是( C )。
- A) i=\*((x+1)+1);

- B)  $i = x[1][1];$
- C)  $i = *(x+1)+1;$
- D)  $i = *(x[1]+1);$

(9) 若定义三维数组 `double a[2][3][4];`则在元素 `a[i][j][k]`前面的元素个数为( B )。  
 A)  $k*2*3+j*2+i$       B)  $i*3*4+j*4+k$     C)  $i*3*4+j*4+k-1$     D)  $k*2*3+j*2+i-1$

(10) 若已定义 `char s[2][40]={"china","English"};` 则在下面表达式中能表示字符'E'的地址的是( B )。  
 A、`&s[2][1]`      B、`s[1]`      C、`*s[1]`      D、`&(s[1]+1)`

(11) 一个 C 程序的执行是从( A )。  
 A)本程序的 `main` 函数开始,到 `main` 函数 `return` 语句结束  
 B)本程序文件的第一个函数开始,到本程序文件的最后一个函数结束  
 C)本程序的 `main` 函数开始,到本程序文件的最后一个函数结束  
 D)本程序文件的第一个函数开始,到本程序 `main` 函数结束

(12) 在 C 语言中,假定一维数组名作为函数调用时的实参,则实际上在函数调用之时传递给函数形参变量的是( C )  
 A) 一维数组的所有元素值      B) 一维数组第一个元素的值  
 C) 一维数组第一个元素的地址      D) 以上都不对

(13) 有程序 `char s[]="ABCD", *P; for(p=s+1; p<s+4; p++) printf ("%s",p);` 程序的输出结果是( D )。  
 A) ABCD      BCD      CD      D  
 B) A      B      C      D  
 C) B      C      D  
 D) BCD      CD      D

(14) 设有定义 `char * lang[]={"FOR", "BAS", "PAS", "JAVA", "C"};`则表达式 `*lang[1]>*lang[3]`的值是( B )。  
 A) 1      B) 0      C) 非零      D)负数

(15) 有以下程序,程序运行后的输出结果是( D )。

```
void main()
{
    int x=3, y=2, z=1;
    printf("%d\n",x/y&~z);
}
```

A) 3      B) 2      C) 1      D) 0

### 三、输出程序运行结果（设程序均已包含所需头文件）（25 分）

评分标准建议：格式写对酌情给分

#### (1) 程序代码块(5')

```
int main()
{
    int a,b,m,n;
    m=n=1;
    scanf("%d%d",&a,&b);
    do
    {   if(a>0)
        {   m=2*n;   b++; }
        else
            {   n=m+n;   a+=2;   b++; }
    }while(a==b);
    printf("m=%d,n=%d\n",m,n);
    return 0;
}
```

从键盘输入-1 0<CR>(<CR>表示回车)，则程序输出结果：

m=4,n=2<CR>

#### (2) 程序代码块(5')

```
int a;
void func(void)
{
    static int b=0;
    auto int c=0;
    printf("a=%d\tb=%d\tc=%d\n",a++ , ++b , c++);
}
int main(void)
{
    func( );
    a=2;
    func( );
    func( );
    return 0;
}
```

程序输出结果：

a=0      b=1      c=0<CR>  
a=2      b=2      c=0<CR>  
a=3      b=3      c=0<CR>

(3) 程序代码块(5')

```
typedef struct Key{
    char *keyword;
    int keyno;
}Map;
void func(Map *PtrSet , int n );
int main()
{
    Map set[]={{"design",123},{ "module",456},{ "test", 789}};
    Map *mPtr=set;
    func(mPtr , 3);
    for( ; mPtr<set+3; mPtr++)
        printf("%s,%d\n", mPtr->keyword, mPtr->keyno);
    return 0;
}

void func(Map *PtrSet , int n )
{
    Map *Ptr=PtrSet;
    for( ;Ptr<PtrSet+n; )
    {
        ++(Ptr->keyno);
        ++(Ptr->keyword);
        ++Ptr;
    }
}
```

程序输出结果:

esign, 124<CR>

odule, 457<CR>

est, 790<CR>

(4) 程序代码块(5')

```

void arrayFunc(char *[] , int );
void main()
{
    char *pstr[6],str[][10]={"python","java","c/c++",
                             "Windows","Linux", "Ubuntu"};

    int    a, b, n = 6;
    for(a=0;a<n;a++)
        pstr[a] =str[a];

    arrayFunc(pstr , 6);
    for (a = 0; a < n; a++)
        printf("%s\n", pstr[a]);
}

```

```

void arrayFunc(char *p2Arr[] , int Rows)
{
    int a,b;
    for (a = 0; a < Rows - 1; a++)
        for (b = 0; b < Rows-a-1; b++)
            if ( strcmp(p2Arr[b], p2Arr[b+1]) > 0 )
            {
                char * temp;
                temp = p2Arr[b];
                p2Arr[b] = p2Arr[b+1];
                p2Arr[b+1] = temp;
            }
}

```

程序输出结果：

Linux<CR>

Ubuntu<CR>

Windows<CR>

c/c++<CR>

java<CR>

python<CR>

(5) 程序代码块(5')

```
char * process_char(char *,char );
int main(void)
{
    char ch[20], ProChar;
    puts("Pls enter a String (no more than 20 characters):");
    fgets(ch,20,stdin);
    puts("\nPls enter a character to process:");
    fflush(stdin); //clear buffers of standard input device
    ProChar =getchar();

    process_char(ch,ProChar);
    puts(ch);
    return 0;
}

char * process_char(char *s,char toProcess)
{
    char *read,*write;
    read=write=s;
    while(*read)
    {
        if(*read==toProcess)    read++;
        else                    *write++ =*read++;
    }
    *write=0;
    return s;
}
```

从键盘输入#1:\$123+#2:\$456<CR>\$<CR>，则程序输出结果：

(直接写出 puts (ch) 执行后的屏幕显示信息)

#1:123+#2:456<CR>

#### 四、程序改错（10 分）

题目要求：不得改变程序框架，不得重写程序，无需文字说明，直接在代码上添加、删除和修改。

(1) 找 N 个字符串中最大字符串和最小字符串。（5'）

```
#include <stdio.h>
#include <string.h>
#define N=5                                #define N 5          0.5 分
char *GetMaxMinString(char (*)[80],int ,char **); 1 分
int main(void)
{
    char string[N][80],*pmax,*pmin;
    int i;

    for(i=0;i<N;i++)
        gets(string[i]);
    pmax =GetMaxMinString(string,N,pmin);
    pmax =GetMaxMinString( string,N, &pmin ); 1 分

    printf("Max string is %s,Min string is %s\n",pmax,pmin);
    return 0; 0.5 分
}

char *GetMaxMinString(char (*str)[80],int num,char ** min )
{
    int a;
    char *max;
    max= str[0];
    *min= str[0];

    for(a=1;a<num;a++)
    {
        if( str[a]>max )      strcmp(str[a],max)>0 1 分
            max =str[a];
        if( str[a]<min )      strcmp(str[a],*min)<0 1 分
            *min =str[a];
    }
    return max;
}
```

第二种方案：用传引用法

```
char *GetMaxMinString(char (*str)[80],int num,char *&min )
{
    ....min= str[0];... if(strcmp(str[a],min)<0)min =str[a];
}
调用 pmax =GetMaxMinString( string,N, pmin );
```

(2) 键盘输入 n 个 0~127 的整数,编程实现将 n 个数按输入时顺序的逆序排列并输出显示



```

#include <stdio.h>
void sort(char *p, int m);           1 分

int main( )
{
    int i,n;
    char *p, num[20];
    scanf("input n: %d", n);      scanf("%d", &n); 1 分
    printf("please input these numbers: \n");
    for (i=0;i<n;i++)
        scanf("%d", &num[i]);
    p = num;
    sort(p, n);
    printf("Now, the sequence is: \n");

    for (i=0;i<n;i++) 去掉分号      0.5 分
        printf("%d", num[i]);
    printf("\n");
    return 0;
}

void sort(char *p, int m)
{
    int i;
    char *temp, *p1, *p2;      char temp;      0.5 分
    for (i=0; i<m; i++)
    {
        p1 = p+i;
        p2 = p + (m-i);      p2 = p + (m-1-i);      1 分
        *temp = *p1;      temp = *p1;      0.5 分
        *p1 = * p2;
        *p2 = *temp;      *p2 = temp;      0.5 分
    }
}

```

## 五、编写程序（35 分，8'+9'+9'+9'）

题目要求：根据题目需要完成代码编写，不得使用全局变量，不得使用 goto 语句，字迹工整，逻辑清晰，程序结构合理。两个标准头文件 `stdio.h` 和 `stdlib.h` 勿用复写，其他头文件若使用，需要自行增加到代码中。可以使用宏定义。

(1) 编写程序：将 3\*3 矩阵 num 做转置操作，输出转置后的结果。（8'）

评分标准：

- 1、没有写出完整的转置操作的循环代码块，扣 4 分；
- 2、其余部分，酌情扣分

```

#include <stdio.h>
#define SIZE 3
int main()
{
    int num[SIZE][SIZE], i, j ;
    for(i=0;i<SIZE;i++)
        for(j=0;j<SIZE;j++)
            scanf("%d",&num[i][j]);
    for(i=0 ;i<SIZE-1;i++)
        for(j=i+1;j<SIZE;j++)
        {
            int temp;
            temp =num[i][j];
            num[i][j] =num[j][i];
            num[j][i] =temp;
        }
    for(i=0 ;i<SIZE;i++){
        for(j=0;j<SIZE;j++)
            printf("%d ",num[i][j]);
        printf("\n");
    }
    return 0;
}

```

- (2) 编写函数 `char *concat(char str1[], char str2[], char add[])` 将两个字符串 `str1` 和 `str2` 连接起来，按照先 `str1` 后 `str2` 次序存储在 `add` 中，在主函数中输入两个字符串 `s1` 和 `s2`，调用 `printf("%s",concat(s1,s2,total))` 打印输出 `total` 中存储的连接串。注意：不能使用库函数 `strcpy`、`strcat`。

```
#include <stdio.h>
char *concat(char str1[], char str2[], char add[]);

int main()
{
    char s1[100], s2[100], total[100];
    printf("input string1:");
    scanf("%s", s1);
    printf("input string2:");
    scanf("%s", s2);
    printf("\nThe new string is")
    printf("%s", concat(s1, s2, total));
    printf("\n");
    return 0;
}

char* concat(char str1[], char str2[], char add[])
{
    int i, j;
    for (i = 0; str1[i] != '\0'; i++)
        add[i] = str1[i];
    for(j = 0; str2[j] != '\0'; j++)
        add[i+j] = str2[j];
    add[i+j] = '\0';
    return add;
}
```

评分标准：

1、main 函数+头文件+可能的原型声明语句，4 分

2、concat 函数，5 分

每一块中出现非主体错误，酌情扣分

- (3) 从键盘输入某班学生某门课的成绩和学号，编程输出最高分及对应学生的学号。

子函数 `int FindMax(int score[],long num[],int n , type MaxNum)` 实现 `n` 个学生的成绩 `score` 和学号 `num` 的统计。要求：1. 函数返回值：找到的最高分；2. 为函数第四个参数 `MaxNum` 选定合适的类型，实现利用 `MaxNum` 把找到最高分的学生学号送回主函数。请给出 `FindMax` 原型声明以及具体实现；主函数完成成绩和学号的录入，以及调用 `FindMax` 函数进行统计，并输出统计结果。(9’)

```
#include <stdio.h>
#define N 30

int FindMax(int score[],long num[],int n,long *MaxNum)
{
    int i;
    int MaxScore = score[0];
    *MaxNum = num[0];
    for (i=1; i<n; i++)
    {
        if (score[i] > MaxScore)
        {
            MaxScore = score[i];
            *MaxNum = num[i];
        }
    }
    return MaxScore ;
}

int main()
{
    int score[N], maxScore;
    int n, i;
    long num[N], maxNum;
    printf("How many students?");
    scanf("%d",&n);
    printf("Input student's ID and score:\n");
    for (i=0; i<n; i++)
    {
        scanf("%d%d", &num[i], &score[i]);
    }
    maxScore =FindMax(score, num,n, &maxNum);
    printf("maxScore = %d, maxNum = %d\n", maxScore, maxNum);
    return 0;
}
```

评分标准:

- 1、int FindMax(int score[],long num[],int n,long \*MaxNum)的参数 4 的类型设计正确得 1 分, main 函数对 FindMax 调用表达式正确得 1 分。其中 long \*和 long & 均为正确的设计方案
- 2、FindMax 函数共 5 分, 主体循环思路正确, 建议至少给 2 分
- 3、main 函数+头文件共 4 分
- 4、可以使用传引用版本设计 int FindMax(int score[],long num[],int n,long &MaxNum);

- (4) 设有 N 个考生, 每个考生的数据包括准考证编号 char ID[20]、姓名 char Name[20]、性别 char Sex 和 4 门功课的成绩 int score[4], 主函数负责输入学生信息以及各单科录取分数线。编程实现: (1) 统计所有学生的总分; (2) 找出至少有一门课低于单科录取线的所有考生信息; (3) 按每个考生的总分从高到低打印排名前 20%的单科超过录取分数线的考生信息;

要求: 考生信息输入、统计每个学生的总分、排序和查找高于单科录取线的考生分别用子函数实现。(9')

评分标准建议:

- 1、结构体类型声明 1 分, main 函数 2 分, 如果没用结构体, 则合并入 main 函数总分
- 2、**statistic** 统计总分给 1 分、**descendsort** 排序 2 分、**search** 查找至少一门课低于录取分数的考生和 **findexceed** 打印单科均超过的前 20%考生信息各 1 分, 如果学生信息输入单独子函数设计给 1 分, 如果没有, 与 main 函数合并 3 分

```
#include <stdio.h>
#define N 6

typedef struct _student
{
    char ID[20];
    char Name[20];
    char Sex;
    int score[4];
    int Total;
} student , *Ptrstudent;

void statistic(Ptrstudent pStu , int n );
void descendsort(Ptrstudent pStu , int n ) ;
void search(Ptrstudent pStu , int n ,int admis[]);
void print(Ptrstudent pStu);
void findexceed(Ptrstudent pStu , int n ,int admis[]);
```

```

int main()
{
    student Stus[N];
    Ptrstudent pStu =Stus;
    int admission[4];

    int i;
    printf("Please enter students' message:\n");
    for( ;pStu<Stus+N;pStu++)
    {
        printf("ID:");
        gets(pStu->ID);
        printf("Name:");
        gets(pStu->Name);
        printf("Sex:");
        fflush(stdin);
        pStu->Sex =getchar();
        printf("score:");
        for(i=0;i<4;i++)
        {
            scanf("%d",&pStu->score[i]);
        }
        fflush(stdin);
    }
    printf("Please enter the admission score of each
                                course:\n");

    for(i=0;i<4;i++)
        scanf("%d",&admission[i]);
    statistic(Stus , N );
    descendsort(Stus , N );
    printf("some courses that score is less than admission
                                line:\n");

    search(Stus , N ,admission);
    findexceed(Stus , N ,admission);
    return 0;
}

```

```

void statistic(Ptrstudent pStu , int n )
{
    int i,j;
    for(i=0 ; i<n ; i++)
    {
        int total=0;
        for(j=0;j<4;j++)
        {
            total +=(pStu+i)->score[j];
            (pStu+i)->Total =total;
        }
    }
}

void search(Ptrstudent pStu , int n ,int admis[])
{
    int i,j;

    printf("%-20s%-20s%4s%-16s%6s\n","ID","Name","sex","score","Total");

    for(i=0 ; i<n ; i++)
    {
        for(j=0 ; j<4; j++)
            if((pStu+i)->score[j]<admis[j])
                break;
        if(j>=4) continue;
        print(pStu+i);
    }
}

void print(Ptrstudent pStu)
{
    printf("%-20s",pStu->ID);
    printf("%-20s",pStu->Name);
    printf("%4c",pStu->Sex);
    for(int i=0 ; i<4; i++)
        printf("%4d",pStu->score[i]);
    printf("%4d\n",pStu->Total);
}

```

```

void findexceed(Ptrstudent pStu , int n ,int admis[])
{
    int i,j;

    printf("%-20s%-20s%4s%-16s%6s\n","ID","Name","sex","score","Total");
    for(i=0 ; i<(int)(n*0.2) ; i++)
    {
        for(j=0 ; j<4; j++)
            if((pStu+i)->score[j]<admis[j])
                break;
        if(j<4) continue;
        print(pStu+i);
    }
}
//降序
void descendsort(Ptrstudent pStu , int n )
{
    int i,j,isModified=0;
    student temp;
    for(i=0;i<N-1;i++)
    {
        isModified =0;
        for(j=0;j<N-i-1;j++)
        {
            if((pStu+j)->Total<(pStu+j+1)->Total)
            {
                temp =(pStu+j);
                *(pStu+j)  =(pStu+j+1);
                *(pStu+j+1)=temp;
                isModified =1;
            }
        }
        if(isModified==0)
            break;
    }
}

```





