

武汉科技大学

2017 年全国硕士研究生招生考试初试自命题试题

科目名称：C 语言程序设计（□A 卷■B 卷）科目代码：855

考试时间：3 小时 满分 150 分

可使用的常用工具：√无 □计算器 □直尺 □圆规（请在工具前打√）

注意：所有答题内容必须写在答题纸上，写在试题或草稿纸上的一律无效；考完后试题随答题纸交回。

一、选择题（共 10 小题，每小题 2 分，共 20 分）

- 下列说法中错误的是（ ）。
A. 主函数可以分为两个部分：主函数说明部分和主函数体
B. 主函数可以调用任何非主函数的其它函数
C. 任何非主函数可以调用其它任何非主函数
D. 程序可以从任何非主函数开始执行
- 求整数 1 至 10 的和并存入变量 s，下列语句中错误的是（ ）。
A. `s=0; for(i=1;i<=10;i++) s+=i;`
B. `s=0;i=1; for(;i<=10;i++) s=s+i;`
C. `for(i=1,s=0;i<=10;s+=i,i=i+1);`
D. `for(i=1;s=0;i<=10,i++) s=s+i;`
- 在 C 语言的语句中，用来决定分支流程的表达式是（ ）表达式。
A. 任意 B. 逻辑 C. 关系 D. 逻辑或关系
- 函数调用语句 `f((e1, e2), (e3, e4, e5))` 中参数个数是（ ）。
A. 5 B. 4 C. 2 D. 1
- 若有定义语句 `int x=10`；则表达式 `x-=x+x` 的值为（ ）。
A. -10 B. -20 C. 0 D. 10
- 以下叙述中正确的是（ ）。
A. 结构体数组名不能作为实参传给函数
B. 结构体变量的地址不能作为实参传给函数
C. 结构体中可以含有指向本结构体的指针成员
D. 即使是同类型的结构体变量，也不能进行整体赋值
- 表达式 `!(x||a==b)` 等效于（ ）。
A. `!((x||a)==b)` B. `!(x||y)==b` C. `!(x||(a==b))` D. `(!x)|| (a==b)`
- 以下叙述中正确的是（ ）。
A. 在 c 语言中，逻辑真值和假值分别对应 1 和 0
B. 关系运算符两边的运算对象可以是 C 语言中任意合法的表达式
C. 对于浮点变量 x 和 y，表达式 `x==y` 是非法的，会出编译错误
D. 分支结构是根据算术表达式的结果来判断流程走向的
- 以下叙述中正确的是（ ）。
A. 只要适当地修改代码，就可以将 for 与 while 相互转换
B. “for(E1; E2; E3) S”先计算 E2 的值，以便决定是否开始循环
C. for 语句只在个别情况下才能转换成 while 语句

D. 如果需要使用无限循环, 则只能使用 while 语句

10. 设有定义: `int x=0, *p;` 紧接着的赋值语句正确的是()。

- A. `*p=NULL;` B. `p=NULL;` C. `p=x;` D. `*p=x;`

二、读程序写出程序运行结果(共 4 小题, 每小题 5 分, 共 20 分)

1. `#include<stdio.h>`

```
void main()
{
    int k=5, n=0;
    do{
        switch(k)
        {
            case 1: case 3: n+=1; k--; break;
            default: n=0; k--;
            case 2: case 4: n+=2; k--; break;
        }
        printf("%d", n);
    }while(k>0&& n<5);
}
```

2. `#include<stdio.h>`

```
#define N 4
void main()
{
    int a[N][N]={0}, i, j, k;
    for(i=0; i<N/2; i++)
    {
        for(j=i; j<N-i; j++) a[i][j]=a[N-i-1][j]=i+1;
        for(k=i+1; k<N-i-1; k++) a[k][i]=a[k][N-i-1]=i+1;
    }
    for(j=0; j<N; j++) printf("%d", a[i][j]);
}
```

3. `#include<stdio.h>`

```
int fun(int a)
{
    int b=2; static int c=2;
    b++; c--; return a+b+c;
}

void main()
{
    int k, a=2;
    for(k=0; k<3; k++) printf("%d", fun(a));
}
```

4. `#include <stdio.h>` //运行时输入: -5 -6 1 7 5 8 2 -4 -3 0

```
void arrange(int p[], int n)
{
    int i=0, j=n-1, t;
    while(i<j)
    {
        while(i<j&& p[i]>0) i++;
        while(i<j&& p[j]<0) j--;
        if(i<j) { t=p[i]; p[i]=p[j]; p[j]=t; }
    }
}

void main()
{
    int a[10], i;
```

```

        for(i=0;i<10;i++)          scanf("%d",&a[i]);
        arrange(a,10);
        for(i=0;i<10;i++)          printf("%d",a[i]);
    }

```

三、程序填空(共 20 个空, 每空 3 分, 共 60 分)

1. 下面程序的功能: 不断从终端读入整数, 分别统计正数和负数的个数, 当输入 0 时结束输入。

```

#include<stdio.h>
void fun(int *pa,int *pb)
{
    ①;    scanf("%d",&k);
    while(②)
    {
        if(k>0) a++;    else b++;
        ③;
    }
    ④;
}

void main(){ int x,y; ⑤; printf("x=%d y=%d\n",x,y); }

```

2. 程序的功能: 将数组中的前半部分元素与后半部分元素对换。若 n 为奇数, 则中间的元素不动。例如, 若 a 数组中的数据为: 1, 2, 3, 4, 5, 6, 7, 8, 9, 则调换后为: 6, 7, 8, 9, 5, 1, 2, 3, 4。数组的元素由键盘输入。

```

#include<stdio.h>
void fun(①)
{
    int i,t,p;
    p=②;
    for(i=0;i<③;i++)
    {
        t=a[i];    a[i]=a[p+i];    ④;
    }
}

void main()
{
    int n,i,a[100];    scanf("%d",&n);
    for(i=0;i<n;i++)    ⑤;
    fun(a,n);
    for(i=0;i<n;i++)    printf("%4d",a[i]);
}

```

3. 下面函数功能: 除了字符串前导的*号之外, 将串中其他*号全部删除。

```

void fun(char *s)
{
    char *p;
    while(①)    s++;
    ②;
    while(*p)
    {
        if(*p!='*') { *s=*p; ③; }
        ④;
    }
    ⑤;
}

```

4. 函数的功能: 将不带头结点的单链表逆置, 即将所有结点从头到尾倒过来,

使得第一个结点变成最后一个结点，第二个结点变成倒数第二个结点，...

```
typedef struct list { int data; struct list *next; } *LinkList;
void fun(①)
{
    LinkList p,s;
    if(head==NULL) ②;
    p=head->next; ③;
    while(p){ ④; p->next=head; ⑤; p=s; }
}
```

四、程序设计(4 小题，前 2 题各 10 分，后 2 题各 15 分，共 50 分)

1. 水仙花数是各位上数字的立方和等于自身的三位数。例如，153 就是水仙花数，因为 $153=1*1*1+5*5*5+3*3*3$ 。设计程序输出所有的水仙花数，每个水仙花数输出占一行，输出格式形为：153=1*1*1+5*5*5+3*3*3
2. 从键盘输入若干整数（0 表示结束），计算所有整数的和。请分别用递归和非递归的方法设计函数 int f（）完成上述任务。
3. 有 2 个带头结点的单链表，链表中每个结点的数据结构描述如下，各结点的数据域递减有序。

```
typedef struct Node { int data; struct Node *next; } *LinkList;
请设计函数 Merge 将两个链表 La 和 Lb 合并，La 和 Lb 仍然保留，生成新的链表 Lc，Lc 各结点的数据域递增有序，并返回 Lc 的结点数。要求，单链表中均无数据域相同的结点，设计的程序运行效率尽量高。
```

```
int Merge(LinkList La,LinkList Lb,LinkList Lc)
```

4. 假如有 n 个打印作业排队等待打印，每个作业都有自己的优先级（用正整数来表示，整数值大的优先级高）。打印机每次会从打印队列中取出排在前面的打印作业，如果在剩下的打印队列中没有比它优先级高的，则打印该作业，否则不会打印该作业，而是将该作业重新添加到打印队列尾部。请设计函数 Print 统计 n 个打印作业中指定打印作业 m 是第几个打印的。

```
int Print(int list[],int n,int m)
```

//list 用来存放队列中每个打印作业的优先级，n 表示等待打印的作业数
//m 为指定的打印作业序号，函数返回值为作业 m 的打印序号。

计算机/软件工程专业

每个学校的

考研真题/复试资料/考研经验

考研资讯/报录比/分数线

免费分享



微信 扫一扫

关注微信公众号

计算机与软件考研

参考答案(B)

一、选择题 (每题 2 分, 共 20 分)

DDACA CDBAB

二、读程序写结果 (每题 5 分, 共 20 分)

1. 235 2. 1221 3. 654 4. 281750-6-4-3-5

三、程序填空题 (20 个空, 每空 3 分, 共 60 分)

- (1) int k, a=0, b=0 (2) k (3) scanf("%d", &k) (4) *pa=a, *pb=b (5) fun(&x, &y)
- (1) int a[], int n (2) (n+1)/2 (3) n/2 (4) a[p+i]=t (5) scanf("%d", &a[i])
- (1) *s== '*' (2) p=s (3) s++ (4) p++ (5) *s=0
- (1) LinkList head (2) return (3) head->next=NULL (4) s=p (5) head=p

四、程序设计题 (4 小题, 前 2 题各 10 分, 后 2 题各 15 分, 共 50 分)

1.

```
#include<stdio.h>
int main()
{
    int i, a, b, c;
    for(i=100; i<1000; i++)
    {
        a=i/100;    b=i/10%10;    c=i%10;
        if(i==a*a*a+b*b*b+c*c*c)
            printf("%d=%d*d*d+%d*d*d+%d*d*d\n", i, a, a, a, b, b, b, c, c, c);
    }
    return 0;
}
```

2.

```
int f() //递归
{
    int x;
    scanf("%d", &x);
    if(x==0) return 0;
    return x+f();
}

int f() //非递归
{
    int s, x;
    s=0;
    scanf("%d", &x);
    while(x) { s=s+x; scanf("%d", &x); }
    return s;
}
```

3.

```
int Merge(LinkList La, LinkList Lb, LinkList Lc)
{
    LinkList p, q, s, t;
    int c=0;
    p=La->next;    q=Lb->next;
    Lc=(LinkList)malloc(sizeof(struct Node));
    Lc->next=NULL;
    while(p&&q)
    {
```

```

s=(LinkedList)malloc(sizeof(struct Node));
if(p->data>q->data) { s->data=p->data; p=p->next; }
else if(p->data<q->data){ s->data=q->data; q=q->next; }
else { s->data=p->data; p=p->next; q=q->next; }
s->next=Lc->next;
Lc->next=s;
c++;
}
if(q) p=q;
while(p)
{
s=(LinkedList)malloc(sizeof(struct Node));
s->data=p->data;
p=p->next;
s->next=Lc->next;
Lc->next=s;
c++;
}
return c;
}
4.
int Print(int list[], int n, int m)
{
int Queue[MaxSize], front, rear; //队列空间、队头和队尾(队列空间足够大, 不会满)
int i, k, t, c=0; //c 用来记载已打印的作业数

front=0; rear=0; //初始化
for(i=0; i<n; i++)
{ Queue[rear]=i; rear=(rear+1)%MaxSize; } //作业顺序入队列
while(front!=rear) //队列不为空
{
k=Queue[front]; front=(front+1)%MaxSize; //出队列
for(j=front; j!=rear; j=(j+1)%MaxSize) //查找队列中有没有比作业k 优先级高的
{ if(list[k]<list[Queue[j]]) break; }
if(j!=rear) { Queue[rear]=k; rear=(rear+1)%MaxSize; } //作业k 放到队尾
else { c++; if(k==m) break; } //作业k 打印, 如果作业k 就是需要找的作业则终止
}
return c;
}
}

```