

上海师范大学 2004 年硕士研究生入学考试试题

专业名称 计算机软件与理论 自然地理学

考试科目 程序设计 (458) C 语言与程序设计 (454)

(注意: 答案必须写在统一印制的答题纸上, 否则不给分)

一、选择题 (20 分, 每题 2 分)

1. 下列程序运行结果是 ()

```
int n=0;
while ( n--<=2) printf("%d",n);
```

- ① 2 ② 3 ③ 4 ④ 有语法错

2. 设 $a=1, b=2, c=3, d=4$, 则表达式: $a++ < b ? a : c < d ? --a : --d$ 的结果为 ()。

- ① 1 ② 2 ③ 3 ④ 4

3. 设有变量定义 $\text{int } a[2][3]$; 数组 a 中元素的地址的下确表示是 ()

- ① $a[1]+3$ ② $*(a+2)$ ③ $*(a[1]+2)$ ④ $*(a+1)$

4. 假定 a 和 b 为 int 型变量, 则执行以下语句后 b 的值为 ()。

```
a=1; b=10;
do { b-=a; a++; } while ( b-- < 0);
```

- ① 8 ② 9 ③ -2 ④ -4

5. 执行以下语句 $a=1, b=c=1$; $--a \&\&--b$ $--c$ 后, 变量 a, b, c 的值依次为 ()

- ① 0 0 0 ② 0 0 1 ③ 0 1 0 ④ 1 0 0

6. 执行下列程序后的输出结果是 ()

```
#include <stdio.h>
int i;
for ( i=1; i<=5; i++ )
switch(i%5)
{ case 0: printf("*"); break;
case 1: printf("#"); break;
default: printf("\n");
case 2: printf("&");
```

- ① $\# \& \& \& *$ ② $\# \&$ ③ $\#$ ④ $\# \&$
- $\&$ $\&$
- $\& *$ $\&$ $*$
- $\&$
- $*$

069

7. 执行下列程序后的输出结果是 ()。

```
#define M(a,b) (a) > (b) ? (a) : (b)
```

```
main ( )
```

```
{ int i=10, j=15; printf ("%d\n", 10*M(i, j)); }
```

① 150

② 100

③ 15

④ 10

8. 设有 `int i = 010;` 则 `printf("%d,%d\n", ++i, i++);` 的输出是 ()。

① 011, 011

② 9, 9

③ 10, 8

④ 11, 11

9. 语句 `printf("%d\n", strlen("a\ts\n\x124\0xb\\"));` 的输出是 ()。

① 11

② 10

③ 9

④ 8

10. 已知在 PC 机上, 定义一数组 `a` 如下, `int a[3][4]={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};` 若数组 `a` 的首地址是 500, 则 `a[1]+3` 的值应是 ()。

① 507

② 508

③ 514

④ 8

二、填空题 (20 分, 每一个空格 2 分)

1. 已知 `static int a[] = { 5, 4, 3, 2, 1};`

```
int p={a+3, a+2, a+1, a}; *q=p;
```

表达式 `*(p[0]+1)+**(&q+2)` 的值为 ①。

2. 下面函数用于求两个非负整数的最大公约数, 请为程序填空。

```
int gcd( int x, int y )
```

```
{ while ( x!=y )
```

```
if (x > y) x = ①; else y = ②;
```

```
return x;
```

3. 下面函数 `strcat(char *s, char *t)` 的功能是将字符串 `s` 连接在字符串 `t` 的后面, 构成一个首指针为 `s` 的字符串。

```
void strcat(char *s, char *t)
```

```
{ while ( *s != '\0' );
```

```
①;
```

```
for ( ; ②; s++, t++ );
```

4. 函数 `sort ()` 将一整数数组 `a[]` 从小到大排序, 请填空。

```
void sort(int *a, int n)
```

```
{ int k, j, start, end, t;
```

```
start=0; end=n-1;
```

```
do
```

```
{ for(k=j=start; j<end; j++)
```

```
if (a[j]>a[j+1])
```

```
t=a[j], a[j]=a[j+1], a[j+1]=t, k=j;
```

```
③=k;
```

```
for( k=j=end; j>start; j-- )
```



```

        if (a[j]>a[j-1])
            t=a[j], a[j]=a[j-1], a[j-1]=t, k=j;
        ② ;
    while ( ③ );
}

```

5. 递归函数 `int max(int a[], int n)` 的返回值是数组 `a[]` 的前 `n` 个元数中的最大值, 请填空。

```

int max(int a[], int n)
{
    int t;
    if ( n == 1 ) return a[0];
    ① ;
    if ( ② ) t=a[n-1];
    return t;
}

```

三、程序分析题 (30 分, 每小题 5 分)

1. 阅读程序, 当输入为 5 时, 写出程序的输出结果。

```

#include <stdio.h>
long fact( int i )
{
    static long k = 1L;
    k=k*i;
    return k;
}

main( )
{
    int n, i, t;
    scanf("%d",&n);
    for ( i=1; i<=n; t=fact(i), i++) ;
    printf("%d!= %ld\n", i, t);
}

```

2. 阅读程序, 写出程序的输出结果。

```

#include <stdio.h>
void sub1(char a, char b) {char c; c=a, a=b, b=c;}
void sub2(char *a, char b) {char c; c=*a, *a=b, b=c;}
void sub3(char *a, char *b) {char c; c=*a, *a=*b, *b=c;}
void sub4(char *a, char *b) {char *c; c=a, a=b, b=c;}
void main()
{
    char a, b;
    a='A'; b='B'; sub4(&a, &b); putchar(a); putchar(b);
    a='A'; b='B'; sub3(&a, &b); putchar(a); putchar(b);
    a='A'; b='B'; sub2(&a, b); putchar(a); putchar(b);
    a='A'; b='B'; sub1(a, b); putchar(a); putchar(b);
    printf("\n");
}

```


071

3. 阅读程序, 写出程序的输出结果。

```

#include <stdio.h>
#include <string.h>
main( )
{
    char a[80]="AB", b[80]="LMNP";
    int i=0;
    strcat(a,b);
    while (a[i++]!='\0') b[i]=a[i];
    puts(b);
}

```

4. 阅读程序, 写出程序的输出结果。

```

#include <stdio.h>
void main( )
{
    char a,b,c,d;
    a=19;
    b=a>>4;
    c=~( b<<4 );
    d=b&c;
    printf("%d,%d,%d,%d\n", a,b,c,d);
}

```

5. 阅读程序, 写出程序的输出结果。

```

#include <stdio.h>
conv( int b)
{
    if ( b>=2 ) conv(b/2);
    printf("%d",b%2);
}
main( )
{
    int d;
    scanf("%d",&d);
    conv(d);
}

```

输入为 22

6. 阅读程序, 写出程序的输出结果。

```

#include <stdio.h>
test( int n)
{
    printf("%4d",n);
    if ( n<=50 )
        test( 2*n-1 ); printf("%4d",n);
    return 0;
}

```



```
main( )
{ test(1); }
```

四、程序设计题 (30 分, 各个空格 4 分)

阅读下列程序说明和 C 程序, 将应填入其中 ① 处的字句, 写在答卷的对应栏内.

1. [程序说明]

在 n 行 n 列矩阵中, 每行都有最大的数, 本程序求这 n 个最大数中的最小一个数.

[程序]

```
#include <stdio.h>
#define N 1000
int a[N][N];
void main( )
{ int row, col, max, min, n;
  /* 输入合法数 n (<100), 和输入 n x n 个整到数组 a 的代码略 */
  for ( row=0; row<n; row++)
  { for ( max=a[row][0], col=1; col<n; col++)
    { if ( ① ) max=a[row][col];
      if ( ② ) min=max;
      else if ( ③ ) min=max;
    }
    printf("the min of max numbers is %d\n", min);
  }
}
```

2. [程序说明]

递归函数 `invert(int a[], int k)` 将指定数组中的前 k 个元素逆置.

[程序]

```
void invert( int a[], int k)
{ int t;
  if ( ① )
  { invert( ② );
    t=a[0], a[0]=a[k-1], a[k-1]=t;
  }
}
```

3. [程序说明]

本程序中的函数 `first_insert()` 的功能是在已知链表的首表元之前插入一个指定值的表元; 函数 `revert_copy()` 的功能是按已知链表复制出一个新链表, 但新链表的表元链接顺序与已知链表的表元链接顺序相反; 函数 `print_link()` 用来输出链表中各表元的值; 函数 `free_link()` 用来释放链表全部表元空间.

[程序]

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


073

```

typedef struct node
{
    int val;
    struct node *next;
} NODE;

void first_insert( NODE **p, int v)
{
    NODE *q=( NODE *)malloc(sizeof(NODE))
    q->val=v; ① ; *p = ② ;
}

NODE *revert_copy( NODE *p)
{
    NODE *u;
    for ( u=NULL;p;p=p->next ) first_insert( ③ );
    return u;
}

void print_link( NODE *p)
{
    for ( ; ④ ) printf("%d\t",p->val);
    printf("\n");
}

void free_link( NODE *p)
{
    NODE *u;
    while (p!=NULL) { u=p->next; free(p); ⑤ ; }
}

void main( )
{
    NODE *link1,*link2;
    int i;
    link1=NULL;
    for ( i=1;i<=10;i++)
        first_insert( &link1,i);
    link2=revert_copy( link1);
    print_link( link1);free_link( link1);
    print_link( link2);free_link( link2);
}

```

4. [程序说明]

本程序用古典的 Erastosthenes 筛选法求从 2 起到指定范围内的素数。如果要找出 2 至 10 中的素数，开始时筛中有 2 到 10 的数，然后取走筛中的最小的数 2，宣布它是素数，并把该素数的倍数都取走。这样，第一步后，筛中还留下奇数 3, 5, 7, 9；重复上述步骤，再取走最小数 3，宣布它为素数，并取走 3 的倍数，于是留下一步 5, 7。反复上述步骤，直到筛中为空时，工作结束。求得 2 至 10 中的全部素数。

程序中用数组 sieve 表示筛子，数组元素 sieve[i] 的值为 1 时，表示 i 在筛子中，值为 -1 时表示数 i 已被取走。

[程序]

```

#include <stdio.h>
#define MAX 22500
main( )
{
    unsigned int i, rang, factor, k;
    int sieve[MAX];
    printf("Please input the rang:");
    scanf("%d", &rang); /* rang 指出在多大的范围内寻找素数 */
    for ( i=2, i<=rang; i++ ) /* 筛子初始化 */
        ① ;
    factor = 2;
    while ( factor <= rang )
    {
        if ( ② ) /* 筛子中最小数是素数 */
        {
            printf("%d\t", factor);
            k=factor;
            while ( k <= rang ) /* 移走素数的倍数 */
            {
                ③ ; /* 从筛子中去掉该数 */
                k= ④ ;
            }
        }
        ⑤ ;
    }
}

```

5. [程序说明]

函数 psort1() 的实现方法是顺序考察 a[] 中元素, 寻找当前考察元数 a[i] 在已确定的不同元素中的插入位置. 如果 a[i] 是一个新的不同元素, 则插入之; 否则, 忽略该元素。

[程序]

```

int psort1(int a[ ], int n)
{
    int i, j, k, low, high, mid, t;
    for (k=1, j=1; i<n; i++)
    {
        low = 0;
        high = ① ;
        while ( low <= high )
        {
            mid=( low + high)/2;
            if ( a[mid] >= a[i] )
                high = mid-1;
            else
                low = mid + 1;
        }
        if ( low >= k ) ②
    }
}

```


075

```

    t = a[i];
    for (j = 3; j >= 1; j--)
        a[j-1] = a[j];
    a[low] = t;
    5 ;
}

return k;
}

int a[] = { 5, 7, 5, 6, 4, 3, 4, 6, 7 };
main ()
{
    int k, n;
    for (k = 0; k < (sizeof a) / sizeof (int); k++)
        printf( "%5d", a[k] );
    printf( "\n\n" );
    n = psort1(a, (sizeof a) / sizeof(int));
    for (k = 0; k < n; k++)
        printf( "%5d", a[k] );
    printf( "\n\n" );
}

```