

## 北京理工大学《C 语言程序设计》期末试题 (A) 卷

### 一、单项选择题 (共 20 分, 每题 2 分)

1、一个 C 程序的执行是从 \_\_\_\_\_。

- A、main() 函数开始, 直到 main() 函数结束。
- B、第一个函数开始, 最后一个函数结束。
- C、第一个语句开始, 最后一个语句结束。
- D、main() 函数开始, 直到最后一个函数。

2、表示关系  $12 \leq x \leq y$  的 C 语言表达式为 \_\_\_\_\_。

- A、 $(12 \leq x) \& (x \leq y)$
- B、 $(12 \leq x) \&\& (x \leq y)$
- C、 $(12 \leq x) | (x \leq y)$
- D、 $(12 \leq x) || (x \leq y)$

3、已知: `int a[4][10]`, 当 " $0 \leq i < 4, 0 \leq j < 10$ " 时, 对数组元素 `a[i][j]` 的引用错误的是 \_\_\_\_\_。

- A、`*( &a[0][0] + 10 * i + j)`
- B、`*( a + i)[j]`
- C、`*( *( a + i) + j)`
- D、`*( a[i] + j)`

4、设有如下定义: `char *aa[2] = { "abcd", "efgh" }`; 则以下说法中正确的是 \_\_\_\_\_。

- A、aa 数组元素的值分别是 "abcd" 和 "efgh"。
- B、aa 是指针变量, 它指向含有两个数组元素的字符型一维数组。
- C、aa 数组的两个元素分别是一个字符串的首地址。
- D、aa 数组的两个元素中各自存放了字符 'a' 和 'e' 的地址。

5、以下循环语句可以正确终止的是 \_\_\_\_\_。

- A、

```
int i = 100;
while( 1 )
{ i = i++ % 100;
if( i > 100 )
break;
}
```
- B、

```
for( ; ; );
```
- C、

```
int k = 32764;
do
{ k++; k++;
}while( k > 0 );
```
- D、

```
int s = 32764;
while( ( s++ % 2 ) || ( s % 2 ) )
s++;
```

6、`sizeof ( float )` 是 \_\_\_\_\_。

- A、一个双精度型表达式
- B、一个整型表达式
- C、一种函数调用
- D、一个不合法的表达式

7、若有以下定义及语句:

```
struct s1
{ char a[3];
int num;
}t = { 'a', 'b', 'c', 4 }, *p=&t;
则输出值为 c 的语句是 _____。
```

- A、`printf( "%c\n", p->t.a[2] );`
- B、`printf( "%c\n", (*p).a[2] );`
- C、`printf( "%c\n", p->a[3] );`
- D、`printf( "%c\n", (*p).t.a[2] );`

8、以下描述不正确的是 \_\_\_\_\_。

- A、在说明子自动变量的时候,就要为其分配存贮单元
- B、在定义函数时需要使用形参,则必须指定形参的数据类型
- C、形参和实参之间可以是值传递
- D、数组名可以作为参数进行传递

9、以下描述正确的是 \_\_\_\_\_。

- A、continue 语句的作用是结束整个循环的执行
- B、只能在循环体内和 switch 语句体内使用 break 语句
- C、在循环体内使用 break 语句或 continue 语句的作用相同
- D、从多层循环嵌套中退出时,只能使用 goto 语句

10、设有如下结构定义:

```
struct jan
```

```
{ int a;
```

```
float b;
```

```
}c2, *p;
```

若有  $p = \&c2$ ; 则对  $c2$  中的成员  $a$  的正确引用是 \_\_\_\_\_。

- A、 $(*p).c2.a$     B、 $(*p).a$     C、 $p \rightarrow c2.a$     D、 $p.c2.a$

## 二、程序填空题 (共 20 分, 每空 2 分)

1、以下程序完成 3 个操作:

(1) 输入 10 个字符串 (每串不多于 9 个字符), 依次存放在数组  $a$  中, 指针数组  $str$  中的每个元素依次指向每个字符串的开始;

(2) 输出每个字符串;

(3) 从这些字符串中选出最小的那个串输出。

请从对应的一组选项中选择正确的选项填空。

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

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

```
main( )
```

```
{ char a[200] = { '\0' }, *str[10], *sp;
```

```
int i, k;
```

```
sp = 【1】 ;
```

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

```
{ scanf( "%s", sp );
```

```
str[i] = sp;
```

```
k = strlen(sp);
```

```
sp += 【2】;
```

```
}
```

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

```
printf( "%s\n", str[i] );
```

```
k = 0;
```

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

```
if( strcmp( str[i], str[k] ) 【3】 )
```

```
k = i;
```

```
printf( "%s\n", str[k] );
```

```
}
```

- 【1】 A、' \0'                      B、str                      C、a                      D、&a[1]  
 【2】 A、1                      B、k                      C、k + 1                      D、k + i  
 【3】 A、> 0                      B、< 0                      C、== 0                      D、!= 0

2、在保存字符串时，对出现在连续位置的若干个字符，如这些字符在 ASCII 表中也是连续出现，则可以将这些字符使用缩写形式进行保存，以节省保存空间。例如用 A-Z 表示 ABCD...XYZ，1-5 表示 12345，c-f 表示 cdef。以下程序是展开输出经过压缩保存的字符串。以下程序仅处理大写英文字母、小写英文字母和数字的同类压缩，即对于形如 A-9，a-C 这样的缩写不进行处理。请从对应的一组选项中选择正确的选项填空。

```
#include <stdio.h>
main()
{ char sa[100];
    int i = 0, k;
    gets( sa );
    do
    { putchar( sa[i] );
      i++;
      if( sa[i] == '-' )
      { k = 1;
        do
        { putchar( 【4】 );
          k++;
        }while( 【5】 );
        i++;
      }
    }while( sa[i] != '\0' );
    printf( "\n" );
}

【4】 A、sa[i-1]                      B、sa[i - 1] + k                      C、sa[i]                      D、sa[i] + k
【5】 A、sa[i-1] + k != sa[i+1]                      B、sa[i] + k != sa[i+1]
      C、sa[i-1] != sa[i+1] + k                      D、sa[i] != sa[i+1] + k
```

3、以下程序将文件 A.DAT 和 B.DAT 合并到文件 C.DAT 中，其中 A.DAT 和 B.DAT 为降序字符文件，保存在当前路径下，文件结束后标志设为 '\0'。请从对应的一组选项中选择正确的选项填空。

```
#include <stdio.h>
main()
{ 【6】 *in1, *in2, *out;
    int flag1 = 1, flag2 = 1;
    char a, b, c;
    if( ( in1 = fopen("A.DAT", "r") ) == NULL )
    { printf( "Can't open the file A.DAT!\n" );
      exit( 1 );
    }
    if( ( in2 = fopen( "B.DAT", "r" ) ) == NULL )
    { printf( "Can't open the file B.DAT!\n" );
```

```

exit( 1 );
}
if( ( out = fopen( "C.DAT",  【7】  ) ) == NULL )
{ printf( "Can't open the file C.DAT!\n" );
exit( 1 );
}
while( !feof( in1 )  【8】  !feof( in2 ) )
{ if( !feof( in1 ) && flag1 )
    a = fgetc( in1 );
    else
        if( feof( in1 ) )
            a =  【9】  ;
        if( !feof( in2 ) && flag2 )
            b = fgetc( in2 );
        else
            if( feof( in2 ) )
                b =  【9】  ;
        if(  【10】  )
        { c = a; flag1 = 1; flag2 = 0; }
        else
        { c = b; flag1 = 0; flag2 = 1; }
        fputc( c, out );
    }
    fclose( in1 );
    fclose( in2 );
    fclose( out );
}

```

- |      |         |          |         |          |
|------|---------|----------|---------|----------|
| 【6】  | A、char  | B、int    | C、float | D、FILE   |
| 【7】  | A、"r"   | B、"w"    | C、"a"   | D、"b"    |
| 【8】  | A、      | B、&&     | C、==    | D、!=     |
| 【9】  | A、0     | B、1      | C、' \n' | D、' \0'  |
| 【10】 | A、a < b | B、a <= b | C、a > b | D、a >= b |

### 三、上机编写程序题（共 40 分）

1、编程实现，输入顶行字符和图形的高，在屏幕上输出如图所示的图形（下例顶行字符为'A'，图形的高为 5）。（10 分）

```

      A
    B C D
  E F G H I
J K L M N O P
Q R S T U V W X Y

```

测试用例：

输入	输出
1. 3 Z	Z

	A B C	
	D E F G H	
2. 1 B	B	
3. 4 M	M	
	N O P	
	Q R S T U	
	V W X Y Z A B	
4. 5 A	A	(公开)
	B C D	
	E F G H I	
	J K L M N O P	
	Q R S T U V W X Y	
5. 0 A		

2、编程实现，任意输入一个偶数  $n$ ，将它们分解为两个素数之和。(10 分)

输出形式为  $* = * + *$ ，例如输入 12，输出：12 = 7 + 5。

测试用例：

输入	输出	
1. 2	2 = 1 + 1	
2. 12	12 = 7 + 5	(公开)
3. 124	124 = 11 + 113	
4. 1234	1234 = 3 + 1231	
5. 20000	20000 = 3 + 19997	

3、编写一个函数，使用递归算法求给定十进制正整数转换为  $M$  进制数 ( $2 \leq M \leq 9$ )。(10 分)

函数的原型： `void itom(int n, int m, char *k)`

其中：参数  $n$  是给定的正整数；参数  $m$  是要求的进制；参数  $k$  是字符指针，当前得到的  $m$  进制数某位的保存位置。转换后的  $m$  进制数保存在外部定义的长度为 16 的字符数组  $sm$  中，该数组最后一个元素，既  $sm[15]$  保存转换后数的最低位，数组空闲的高位用数字字符 '0' 填充，例如整数 11 转换为二进制后，数组  $sm$  的内容为：

{ '0', '0', '0', '0', '0', '0', '0', '0', '1', '0', '1', '1' }。

输出结构为：1011。

函数没有返回值。

注意：所提交的函数中，在变量说明语句后，其他可执行语句前写 TEST 四个大写字母。此四个大写英文字母必须写在正确的位置，否则此题不能得分。

预设代码：

```
#include <stdio.h>
#define TEST printf( "%s = %c \n", *s )
char sm[17];
main( )
{ int i, n, m;
```

```

for( i = 0; i < 16; i++)
    sm[i] = '0';
printf( "\nPlease enter n m:" );
scanf( "%d %d", &n, &m);
itom( n, m, &sm[15] );
for( i = 0; i < 16; i++ )
if( sm[i] != '0' )
break;
    while( sm[i] != '\0' )
    {   putchar( sm[i] );
        i++;
    }
    printf( "\n" );
}

```

参考答案:

```

itom(int n, int m, char *s)
{   TEST;
    if( n > 0 )
    {   *s = (n % m) + 48;
        itom( n/m, m, s-1);
    }
    else
        return;
}

```

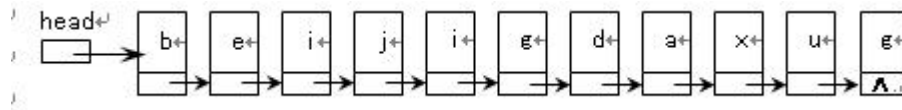
测试用例:

输入	输出	
1. 11 2✓	(5) 1011✓	(公开)
2. 12 2✓	(5) 1100✓	(公开)
3. 12 3✓	(4) 110✓	
4. 12 4✓	(3) 30✓	
5. 12 5✓	(4) 22✓	
6. 527 6✓	(5) 2236✓	(公开)
7. 527 7✓	(5) 1132✓	
8. 527 8✓	(5) 1017✓	
9. 527 9✓	(4) 645✓	
10. 527 2✓	(11) 1000001111✓	

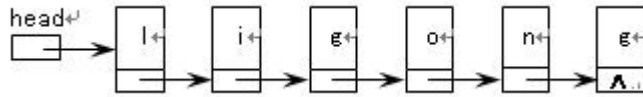
(注: 括号内是在输出结果前输出 “\*s = 0”的次数)

4、用单向链表的方式保存输入的字符串，每一个结点的数值域保存一个字符。程序的功能将输入的两个字符串保存为链表 A 和链表 B，将链表 B 插入到链表 A 的第 n 个结点的后面；如果链表 A 的长度小于 n，则链表 B 接于链表 A 尾。预设的代码包括主函数、建立链表函数、输出链表函数，请编写插入链表的函数。

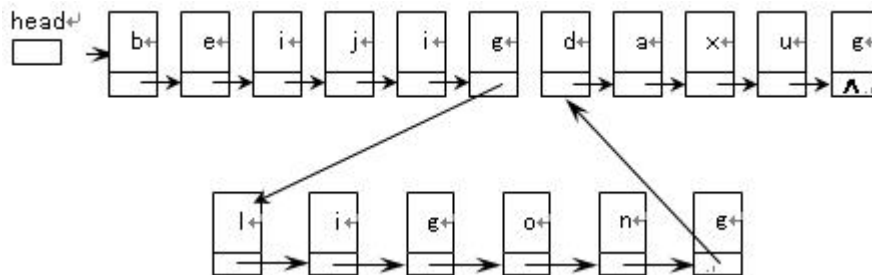
例如以链表方式保存的字符串 “beijingdaxue” 如下图：



字符串“ligong”如下图：



输入 n = 6 后，合并后的字符串如下图：



结构定义是：

```
typedef struct sdata
{   char ch;
    struct sdata *next;
}SNODE;
```

函数原型是：

```
SNODE *inststr(SNODE *heada, SNODE *headb, int m)
```

其中 heada、headb 是保存字符串的链表的头指针，m 是链表 heada 的插入位置。

预设代码如下：

```
#include <stdio.h>
SNODE *setlink(char *str)
{   char *ps = str;
    SNODE *head = NULL, *q, *p = NULL;
    while( *ps != '\0' )
    {   q = ( SNODE *)malloc(sizeof( SNODE ));
        q->ch = *ps;
        q->next = NULL;
        if( head == NULL )
            head = q;
        else
            p->next = q;
        p = q;
        ps++;
    } return( head );
}
SNODE *outlink( SNODE *head )
```

```
{ while( head != NULL )
    { printf("%c",head->ch);
head = head->next;
}
return( head );
}
main( )
{ char sa[100],sb[100];
  int i,n;
  SNODE *head1 = NULL, *head2 = NULL ;
  gets( sa );
  gets( sb );
  scanf( "%d", &n );
  if( *sa != '\0' && *sb != '\0' )
  { head1 = setlink( sa );
    head2 = setlink( sb );
head1 = inststr( head1, head2, n );
    outlink( head1 );
  }
  printf( "\n" );
}

参考答案：
SNODE *inststr(SNODE *heada, SNODE *headb, int n )
{ int i;
  SNODE *q = heada, *p = NULL;
  if( n > 0 )
  { for(i=1; i<n && q->next != NULL; i++)
      q = q->next;
    p = q->next;
    q->next = headb;
    q = headb;
    while( q->next != NULL )
        q = q->next;
    if( p != NULL )
        q->next = p;
  }
  else
  { q = headb;
    while( q->next != NULL )
        q = q->next;
    q->next = heada;
    heada = headb;
  }
  return(heada);
}
```



测试用例：

输入	输出
1. beijingdaxue↵ ligong↵ 7↵	beijingligongdaxue↵ (公开)
2. beijing↵ daxue↵ 7↵	beijingdaxue↵
3. ab↵ cd↵ 5↵	abcd↵ (公开)
4. abcd↵ 12↵ 1↵	a12cdb↵
5. abcd↵ 12↵ 2↵	ab12cd↵
6. abcd↵ 12↵ 3↵	abc12d↵
7. abcd↵ 12↵ 0↵	12abcd↵
8. abcd↵ ↵ 2↵	↵ (公开)
9. ↵ abcd↵ 1↵	↵
10. ↵ ↵ 2↵	↵

参考答案：

- 一、A B B D/C C B B A B B  
二、C C B B A D B A D C