厦门大学《C语言程序设计》课程试卷

[redacted]

一．选择题(30 分，每题 2 分)

1. 关于 C 程序的叙述，错误的说法是(　B　)。
   A. C 程序总是从主函数开始执行　　B. C 程序中定义的第一个函数是主函数
   C. 在主函数中可以调用其它函数　　D. 一个 C 程序可以包括多个函数

2. 下列关于单目运算符++的叙述中正确的是(　D　)。
   A. 它们的运算对象可以是任意变量和常量
   B. 它们的运算对象可以是 char 型变量和 int 型变量，但不能是 float 型变量
   C. 它们的运算对象可以是 int 型变量，但不能是 double 型变量和 float 型变量
   D. 它们的运算对象可以是 char 型变量、int 型变量和 float 型变量

3. 对于代数表达式$\frac{4cd}{ab}$，错误的 C 语言表达式为(　D　)。
   A. 4*c*d/a/b　　B. c*d/a/b*4
   C. c/a/b*d*4　　D. 4*c*d/a*b

4. 以下选项中值为 1 的表达式是(　B　)。
   A. 1 - '0'　　B. 1 - '\0'　　C. '1' - 0　　D. '\0' - '0'

5. 当用户要求输入的字符串中含有空格时，应使用的输入函数是(　C　)。
   A. scanf( )　　B. getchar( )　　C. gets( )　　D. getc

6. 以下 4 个选项中，不能看作一条语句的是(　C　)。
   A. ;　　B. a=5, b=6, c=7;
   C. if (b!=5) x=2; y=3;　　D. return j;

7. 如果整型量 a, b, c 的值分别是 5、4、3，则语句 if (a>b>c) c++; 执行后 c 的值是(　C　)。
   A. 语法错误　　B. 4　　C. 3　　D. 2

8. 已知程序段 int x=0; while (x=1) { … }，则以下叙述正确的是(　C　)。
   A. 循环控制表达式不正确　　B. 循环控制表达式的值为 0
   C. 循环控制表达式的值为 1　　D. 以上说法都不对

9. 当全局变量与函数内部的局部变量同名时， 则在函数内部(　A　)。

A. 局部变量有效, 全局变量被暂时屏蔽

B. 全局变量有效, 局部变量无效

C. 全局变量与局部变量都有效, 全局变量的值被局部变量修改

D. 全局变量和局部变量不能同名

10. 设有函数定义: void p(int x) { printf("%d", x); }, 则正确的函数调用是(　A　)。

A. p(3);　　B. a=p(3);　　C. printf("%d"; p(3));　　D. p(int x);

11. 正确的定义是(　B　)。

A. int a[ ]={ "abcdef" };　　B. int a[ ]={ 1, 2, 3, 4, 5 };

C. char a="y";　　D. int a[3][ ]={ 0, 1, 2, 3, 4, 5 };

12. 有以下语句: char b[8]; int c; , 则正确的输入语句是(　D　)。

A. scanf("%s%d", &b, &c);　　B. scanf("%s%d", &b, c);

C. scanf("%s%d", b, c);　　D. scanf("%s%d", b, &c);

13. 若有定义 int a[10], *p=a; , 则 p+3 表示(　B　)。

A. 数组元素 a[3]的值　　B. 数组元素 a[3]的地址

C. 数组元素 a[4]的地址　　D. 数组元素 a[0]的值加上 3

14. 设有定义 int a[10], *p=a; 下列对数组元素 a[1]的引用中, 错误的是(　B　)。

A. p[1]　　B. *(++a)　　C. *(++ p)　　D. *(a+1)

15. 设有以下说明语句:

```
struct ex
{ int x; float y; char z; } example;
```

则下面的叙述中不正确的是(　B　)。

A. struct 是结构体类型的关键字　　B. example 是结构体类型名

C. x, y, z 都是结构体成员名　　D. struct ex 是结构体类型名

二. 填空题(30 分, 每题 2 分)

1. 设有定义: int a=3; , 则语句 printf("%d,%d\n", a==3, a=3); 的输出结果为__1, 3__。

2. 以下语句的执行结果是_____16_____。

```
int m=0, s=1; do { s+=m++; } while (m<6); printf("%d\n", s);
```

3. 当 a=1, b=2, c=3 时，执行下面语句后, a, b, c 的值分别为 3 、 2 、 2 。

```
if  (a>c)  b=c;  a=c;  c=b;
```

4. 执行语句 x=(a=10, b=a--)后, x、a、b 的值依次为 10 、 9 、 10 。

5. 下列条件语句对应的条件表达式为 x>0 ? 1: (x<0 ? -1: 0) 。

```
if  (x>0)  y=1;  else if  (x<0)  y=-1;  else  y=0;
```

6. 下列程序运行后的输出结果是 2 3 3 。

```
void main( )
{    int i=1, j=2, k=3;
     if  (i++==1 && (++j==3 || k++==3))    printf("%d %d %d\n", i, j, k);    }
```

7. 以下程序段中的变量已正确定义，则程序段的输出结果是 * 。

```
for  (i=0;  i<4;  i++, i++ )   for  (j=1;  j<3;  j++) ;  printf("*");
```

8. 定义如下变量和数组：`int  i, x[3][3]={  1, 2, 3, 4, 5, 6, 7, 8, 9  };`
则下面语句的输出结果是 7 5 3 。

```
for  (i=0;  i<3;  i++)  printf("  %d", x[2-i][i]);
```

9. 写出以下程序段的运行结果 1 , 1 3 , 1 3 , 3 。

```
int  i,  j;
for  (i=1; i<4; i++)
{    for  (j=0;  j<=i;  j++)
     {   if  (j%2==0)    continue;
         if  (i%2==0)    break;
         printf("%d, %d  ", i,  j);
     }
}
```

10. 下面的程序段运行后, x 的值是 0 。

```
#include <stdio.h>
void  main( )
{    int i, j, x=0;
     static int  a[8][8];
     for  (i=0;  i<3;  i++)
          for  (j=0;  j<3;  j++)
               a[i][j]=2*i+j;
     for  (i=0;  i<8;  i++)    x+=a[i][j];
}
```

11. 有以下程序:

```
#include <stdio.h>
int fun(int a, int b) { if (b==0) return a; else return fun(--a, --b); }
void main( ) { printf("%d\n", fun(4, 2)); }
```

程序的运行结果是 ____2____ 。

12. 运行以下程序, 输出结果为 ___5 6___ 。

```
int fun(int a, int *b)
{ a++; (*b)++; return a+*b; }
void main( )
{ int x=1, y=2; printf("%d ", fun(x, &y)); printf("%d ", fun(x, &y)); }
```

13. 运行以下程序, 输出结果为 ___13___ 。

```
int fun(char *x)
{ char *y=x; while (*y) y++; return y-x; }
void main( )
{ char x[ ]="I love China!"; printf("%d\n", fun(x)); }
```

14. 有以下程序:

```
#include <stdio.h>
void point(char *p)
{ p+=3; printf("%c ", *p); }
void main( )
{ char a[ ]={ '1', '2', '3', '4' }, *p=a; point(p); printf("%c\n", *p); }
```

程序运行后的输出结果是 ___4 1___ 。

15. 有以下程序:

```
void main( )
{
    int a[ ]={ 11, 13, 14, 15, 16, 17, 18 }, i=0, j=0;
    while (i<7 && a[i]%2) { j+=a[i]; i++; }
    printf("%d\n", j);
}
```

执行后输出结果是 ___24___ 。

三．程序设计题(共 40 分) 注意：程序中请添加必要的注释

1. 编写程序，输入一个正整数，然后从个位开始一次输出每一位数字对应的英文字母。例如：输入 3927，输出 seven two nine three。（10 分）

```
#include <stdio.h>
#include <string.h>
void  main( )
{
    int  n, i;
    char  a[100]="",  t[100]="";
   //a 数组用于存放已处理过的数字所对应的英文字符串，t 数组用于存放当前所处理的数字对应的英文字符串
    printf("请输入一个正整数： ");
    scanf("%d", &n);
    printf("%d 为： ", n);
    while (n)
    {
        i=n%10;  n=n/10;
        switch (i)
        {    //将当前处理的数字所对应的英文字符串拷贝到字符数组 t 中
            case 0:  strcpy(t, " zero ");  break;
            case 1:  strcpy(t, " one ");  break;
            case 2:  strcpy(t, " two ");  break;
            case 3:  strcpy(t, " three "); break;
            case 4:  strcpy(t, " four ");  break;
            case 5:  strcpy(t, " five ");  break;
            case 6:  strcpy(t, " six ");  break;
            case 7:  strcpy(t, " seven ");  break;
            case 8:  strcpy(t, " eight ");  break;
            default:  strcpy(t, " nine ");  break;
        }
        strcat(t, a);    //a 拼接在 t 后面
        strcpy(a, "");  //a 重新置空
        strcat(a, t);    //t 拼接在 a 后面
    }
    puts(t);
}
```

2. 编写程序，对一个已经排好序的一维整数数组，要求：插入 1 个新的整数后，仍保持原来的排序，然后将新的数组输出。（15 分）

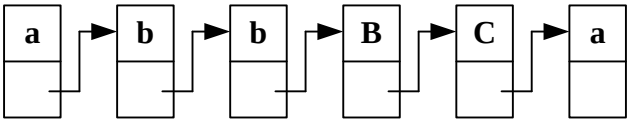
```
#include <stdio.h>
void  main( )
{
      int  a[100],  x,  i=0,  j,  k;
      //a 数组用于存放排序好的整数序列, x 存放当前输入的一个整数。
      printf("请输入已排序的数列，以-1 结束\n");
      scanf("%d", &x);
      while  (x!=-1)    //建立排序序列
      {
          a[i++]=x;
          scanf("%d", &x);
      }
      printf("请输入要插入的数据：");
      scanf("%d", &x);
      if (x<=a[0])          //判断插入的整数是否应该排在序列起始位置
      {
          for (j=i-1; j>=0; j--)
                a[j+1]=a[j];    //序列整体向后移动一个数据位置
          a[0]=x;
      }
      else if  (x>=a[i-1])      a[i]=x;      //判断插入的整数是否应该排在序列末尾位置
      else
      {       for (j=0; j<i-1; j++)
              {
                  if (x>=a[j] && x<=a[j+1])    //确定插入数据所在序列的位置 j+1
                  {
                      for (k=i-1; k>=j+1; k--)
                          a[k+1]=a[k];  //序列整体从 j+1 向后移动一个数据位置
                      break;
                  }
              }
              a[j+1]=x;
      }
      for (j=0; j<=i; j++)
              printf("%d ", a[j]);
      putchar('\n');
}
```

3. 输入一个字符串，请编写一个函数统计连续相同字符及其数量。例如，输入“aabbbbcccAB”，返回“a2 b4 c3 A1 B1”及 5；又如，输入“ABccBBBAc”，返回“A1 B1 c2 B3 A1 c1”和 6。要求：所编写的函数中应采用指针法来引用输入字符串中的元素。（15 分）

例如，对于输入的“abbBCa”字符串，

```
struct   ListNode
{   char   data;
    struct   ListNode   *next;
};
```

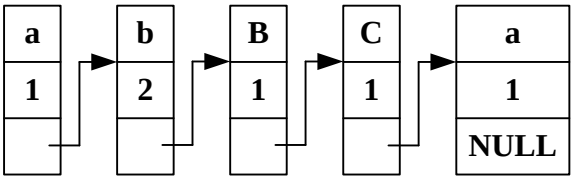
结点结构链表如下：



根据

```
struct   ResultNode
    {      char   data;   int   num;   //num 存放 data 字符出现的个数
           struct   ResultNode   *next;
    };
```

结点结构，对应的输出结果链表如下：




```
#include <stdio.h>
#include <string.h>
#define   LEN   sizeof(struct ListNode)   // 定义 ListNode 结点存储空间大小
#define   SLEN   sizeof(struct ResultNode)   // 定义 ResultNode 结点存储空间大小
struct   ListNode          //输入串字符结点结构
{      char   data;
       struct   ListNode   *next;
};
struct   ResultNode        //输出串字符结点结构, num 存放 data 字符出现的个数
{      char   data;
       int   num;
       struct   ResultNode   *next;
};
struct   ListNode   *createlist( );        //建立输入字符串结构链表
struct   ResultNode   *stat(struct   ListNode   *head,   int *n);
// stat 函数统计字符个数, n 是 ResultNode 结构链表中结点个数的指针
void   printlist( struct   ResultNode   *head );       //输出字符结构链表
```

```
void  main(  )
{
    struct  ListNode  *head;
    struct  ResultNode  *head1;
    int  n=0;
    head = createlist( );  //创建字符串结构链表, 返回头结点指针到 head
    head1 = stat(head, &n);   //统计字符串中连续字符的个数, 返回头结点指针到 head1
    printlist(head1); //输出字符串中连续字符的个数的统计结果
    printf("%d\n", n );
}
void  printlist( struct  ResultNode  *head )
{
    struct  ResultNode *p = head;
    while  (p)
    {
        printf("%c",  p->data);
        printf("%d ", p->num );
        p = p->next;
    }
    printf("\n");
}
struct  ListNode  *createlist(  )
{
    struct ListNode *h, *p1, *p2;
    char s[100];          //存放输入串
    int i;
    p1=(struct ListNode *)malloc(LEN);
    printf("请输入字符串:");
    gets(s);
    p1->data=s[0];   //存放第一个字符
    h=p1;
    for (i=1;  i<strlen(s);  i++)  //处理存放第二个~最后一个字符
    {
        p2=(struct ListNode *)malloc(LEN);
        p2->data=s[i];
        p1->next=p2;
        p1=p2;
    }
    p1->next=NULL;
    return h;
```

```
}
struct ResultNode *stat(struct ListNode *head, int *n)
{       struct    ListNode *p=head;
        struct    ResultNode *h, *p1, *p2;
        char    a;     //存放前一个字符
        int    i=0;     //累计每种字符的个数
        p1=(struct ResultNode *)malloc(SLEN);              h=p1;
        while (p)       //当 p 指向'\0'时，循环结束
        {     a=p->data;          i++;          p=p->next;     //p 指向当前处理字符
              if    (p)     //判断是否处理到'\0'字符
              {       if    (p->data != a)     //比较前一字符和当前字符
                      {
                              (*n)++;
                              p1->data=a;
                              p1->num=i;
                              i=0;
                              p2=(struct ResultNode *)malloc(SLEN);
                              p1->next=p2;              p1=p2;       //p1:当前字符结点指针，p2:下一字
                                                                     符结点指针
                      }
              }
              else        //此时, p1 是'\0'字符的前一个字符
              {         (*n)++;             p1->data=a;             p1->num=i;          }
        }
        p1->next=NULL;
        return h;
}
```