

考试科目：计算机技术

科目代码：853#

适用专业：生物医学工程

(试题共 6 页)

(答案必须写在答题纸上, 写在试题上不给分)

## 一、选择题(40 分, 每题 2 分)

1. 已知字母 A 的 ASCII 码为十进制的 65, 则下面程序的输出是\_\_\_\_\_

```
main()
{char ch1,ch2;
  ch1='A'+5-'3';
  ch2='A'+6-'3';
  printf("%d,%c\n",ch1,ch2);
}
```

A. 67,D      B. B.C      C. C,D      D. 不确定的值

2. 若有代数式:  $\frac{3ae}{bc}$ , 则不正确的 C 语言表达式为\_\_\_\_\_

A. a/b/c\*e\*3      B. 3\*a\*e.b/c      C. 3\*a\*e/b\*c      D. a\*e/b/c\*3

3. 程序运行结果是\_\_\_\_\_

```
main()
{int n=0;
  while(n<=2)
  {n++;
   printf("%d",n);
  }
}
```

A. 1      B. 12      C. 123      D. 1234

4. 设有以下语句:

```
struct st
{int n;
  struct st *next;
};
struct st a[3]={5,&a[1],7,&a[2],9,'\0'}.*p;
p=&a[0];
```

表达式(++p)->n 的值为\_\_\_\_\_

A. 5      B. 7      C. 9      D. '\0'

5. 执行下列语句中, sum 变量的值是\_\_\_\_\_

```
int sum=0;
for(int i=0;i<10;i++,sum+=i);
```

A. 45      B. 55      C. 0      D. 编译错误

6. 已知有共用体变量 data1 定义如下:

```
union data
```

```
{ int i;
```

```
  char ch;
```

```
  float f;
```

```
} data1; 则变量 data1 所占的内存存储空间可表示为
```

A. sizeof(int)

B. sizeof(char)

C. sizeof(float)

D. sizeof(int)+sizeof(char)+sizeof(float)

7. 若 fp 是指向某文件的指针, 且 feof(fp) 的返回值是非零值, 则表示

A. 已读到此文件末尾

B. 未读到此文件的末尾

C. 此文件已关闭

D. 此文件尚未被打开

8. 以下 C 语言函数声明中, 不正确的是\_\_\_\_\_

A. void fun (int x, int y);

B. fun (int x, int y);

C. int fun (int x,y);

D. char \*fun (char \*s);

9. 下列选项中, 不合法的 C 语言关键字是\_\_\_\_\_

A. auto

B. default

C. static

D. var

10. 若 x=4, 则 x\*=x+2 的值为\_\_\_\_\_

A. 36

B. 24

C. 18

D. 20

11. 有以下程序

```
main()
```

```
{  int m=12,n=34;
```

```
    printf("%2d%2d",m++,++n);
```

```
    printf("%2d%2d\n",n++,++m);
```

```
}
```

程序运行后的输出结果是\_\_\_\_\_

A. 12353514

B. 12353513

C. 12343514

D. 12343513

12. 有定义语句: int b; char c[10];, 则正确的输入语句是\_\_\_\_\_

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

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

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

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

13. 有以下程序

```
main()
```

```
{  int m,n,p;
```

```
    scanf("m=%dn=%dp=%d",&m,&n,&p);
```

```
    printf("%d%d%d\n",m,n,p);
```

```
}
```

若想从键盘上输入数据, 使变量 m 中的值为 123, n 中的值为 456, p 中的值为 789, 则正确的输入是\_\_\_\_\_

A. m=123n=456p=789

B. m=123 n=456 p=789

C. m=123,n=456,p=789

D. 123 456 789

14. 有以下程序

```
main()
```

```
{
```

```
    int a,b,d=25;
```

```
    a=d/10%9;
```

```
    b=a&&(-1);
```

```
    printf("%d,%d\n",a,b);
```

```
}
```

程序运行后的输出结果是\_\_\_\_\_

- A. 6,1      B. 2,1      C. 6,0      D. 2,0

15.有以下程序

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

程序运行后的输出结果是\_\_\_\_\_

- A. 1 2 3      B. 2 3 4      C. 2 2 3      D. 2 3 3

16.若整型变量a、b、c、d中的值依次为：1、4、3、2。则条件表达式a<b?a:c<d?c:d的值\_\_\_\_\_

- A. 1      B. 2      C. 3      D. 4

17.有以下程序

```
main()
{
    int p[8]={11,12,13,14,15,16,17,18},i=0,j=0;
    while(i++<7) if(p[i]%2) j+=p[i];
    printf("%d\n",j);
}
```

程序运行后的输出结果是\_\_\_\_\_

- A. 42      B. 45      C. 56      D. 60

18.有以下程序

```
main()
{
    char a[7]="a0\0a0\0";int i,j;
    i=sizeof(a); j=strlen(a);
    printf("%d %d\n",i,j);
}
```

程序运行后的输出结果是\_\_\_\_\_

- A. 2 2      B. 7 6      C. 7 2      D. 6 2

19.以下能正确定义一维数组的选项是\_\_\_\_\_

- A) int a[5]={0,1,2,3,4,5};  
B) char a[]={0,1,2,3,4,5};  
C) char a={'A','B','C'};  
D) int a[5]="0123";

20.有以下程序

```
int f1(int x,int y){return x>y?x:y;}
int f2(int x,int y){return x>y?y:x;}
main()
{
    int a=4,b=3,c=5,d=2,e,f,g;
    e=f2(f1(a,b),f1(c,d));
    f=f1(f2(a,b),f2(c,d));
    g=a+b+c+d-e-f;
    printf("%d.%d%d\n",e,f,g);
}
```

程序运行后的输出结果是\_\_\_\_\_

- A. 4, 3, 7      B. 3, 4, 7      C. 5, 2, 7      D. 2, 5, 7

二、完善程序：请在以下程序的空白处填上正确的内容(30分，每空2分)

1. 以下程序中函数 huiwen 的功能是检查一个字符串是否是回文，当字符串是回文时，函数返回字符串：yes!，否则函数返回字符串：no!，并在主函数中输出。所谓回文即正向与反向的拼写都一样，例如：adgda。

```
#include "string.h"
char *huiwen(char *str)
{
    char *p1,*p2; int i,t=0;
    p1=str; p2= [1]
    for(i=0;i<=strlen(str)/2;i++)
        if(*p1++!=*p2--)
            { t=1; break; }
    if([2]) return ("yes!");
    else return ("no!");
}
main()
{
    char str[50];
    printf("Input:"); scanf("%s",str);
    printf("%s\n",[3]);
}
```

2. 下面程序的功能是用函数递归方法计算菲波那契数列。

```
main()
{ int n,m;
    scanf("%d",&n);
    [1];
    printf("%d",m);
}
int fibo(n);
    int n;
{ if([2]) return(0);
    else if(n==2) return(1);
    else return ([3]);
}
```

3. 下列程序将 HELLO!反向打印出来。

```
#include <stdio.h>
main()
{printf("\nHELLO!\n");
Print(char *string)
{ char *str= [1];
    if(*str= [2]) return
    while(*str) str++;
    [3];
    printf("[4]", *str);
    [5]='\\0';
    print [6];
}
```

4. 以下程序用于统计字符串中最长中词的长度和该单词在字符串中的位置, 其中单词全由字母组成。

```
int alph(c)
char c;
{if ((c>='a'&&c<'z')||(c>='A'&&c<'Z')) return(1);
 else return(0);
}

main()
{static char string[]={"I am happy"};
 int len=0,i,length=0,flag=0,p;
 for(i=0; string[i]!='\0'; i++)
     if(alph(string[i]))
         if(flag)
             {p=i;
              [1];
              len++;
             }
         else
             [2]
     else
         { flag=1;
           if(len>length)
               length=len;
           [3]
         }
 printf("%d, %d", length,p);
}
```

- 三、一个班有 4 名学生, 每个学生学 4 门课, 要求编写程序输出第  $n$  个学生的全部成绩。(10 分)

- 四、输入整数  $n$ , 求表达式

$2^0 + (2^0 + 2^1) + (2^0 + 2^1 + 2^2) + \dots + (2^0 + 2^1 + \dots + 2^n)$  的值。要求在程序中求  $2^x$  的功能部分必须在一函数中完成, 函数名使用 `exp2`。(10 分)

- 五、求出 10 至 1000 之内能同时被 2、3、7 整除的数, 并输出。(10 分)

- 六、用一个一维数组  $S$  (设大小为 `MAX`) 作为两个栈的共享中间。请说明共享方法, 栈满、栈空的判断条件, 并用 C 语言设计公用的入栈操作 `Push(i, x)`, 其中  $i$  为 0 或 1, 用于表示栈号,  $x$  为入栈元素。(10 分)

- 七、设有一个表头为 `first` 的单链表。试设计一个算法, 通过遍历一次链表, 将链表中所有结点按逆序链接。(10 分)

- 八、已知二叉树中的结点类型用 `BinTreeNode` 表示, 定义为:

```
struct BinTreeNode{char data; BinTreeNode *leftChild, *rightChild;;};
```

其中 `data` 为结点值域, `leftChild` 和 `rightChild` 分别为指向左、右子女结点的指针域, 根据下面函数声明编写出求一棵二叉树高度的算法, 该高度由函数返回。假定根结点的层次为 0, 参数 `BT` 初始指向这棵二叉树的根结点。

```
int BtreeHeight (BinTreeNode * BT); (10 分)
```

九、说明下面递归过程的功能。(10分)

```
void unknown(BinTreeNode *T,int a[],int i){  
    if(T!=NULL){  
        a[i]=T->data;  
        unknown(T->leftChild,a,2*i+1);  
        unknown(T->rightChild,a,2*i+2);  
    }  
}
```

主程序调用方式 unknown(BT.root,a,0);

十、设  $n$  是偶数，试计算运行下列程序段后  $m$  的地址并给出该程序段的时间复杂度。  
(10分)

```
m=0;  
for (i=1; j<=n; i++)  
    for(j=2*i; j<=n; j++)  
        m=m+1;
```