

# 中国海洋大学 2020 年硕士研究生招生考试试题

科目代码： 977

科目名称： 高级程序设计与软件工程

## 一、填空题 ( 共 10 题 , 每题 2 分 , 共 20 分 )

1、爱因斯坦的阶梯问题：设有一阶梯，每步跨 2 阶，最后余 1 阶；每步跨 3 阶，最后余 2 阶；每步跨 5 阶，最后余 4 阶；每步跨 6 阶，最后余 5 阶；只有每步跨 7 阶时，正好到阶梯顶。问共有多少阶梯。

```
void main()
{int ladders=7;
  while(【  】) ladders+=14;
  printf("%d\n",ladders);
}
```

2、以下程序的输出结果是【 】。

```
void amovep(int *p, int (*a)[3],int n)
{ int i,j;
  for(i=0;i<n;i++)
    for(j=0;j<n;j++){ *p=a[i][j]; p++; }
}
void main()
{ int *p,a[3][3]={{1,3,5},{2,4,6}};
  p=(int*)malloc(100);
  amovep(p,a,3);
  printf("%d%d\n",p[2],p[5]);free(p);
}
```

3、以下程序的输出结果是【 】。

```
void main()
{ int b[3][3]={0,1,2,0,1,2,0,1,2},i,j,t=1;
  for(i=0;i<3;i++)
    for(j=0;j<=i;j++) t=t+b[i][b[i][j]];
  printf("%d\n",t);
}
```

4、以下程序的输出结果是【 】

---

特别提醒：答案必须写在答题纸上，若写在试卷或草稿纸上无效。

```

int x=3;
void main()
{ int i;
  for(i=1;i<x;i++) incre();
}
void incre()
{ static int x=1;
  x*=x+1;
  printf(" %d",x);
}。

```

5、以下程序运行后的输出结果是【 】。

```

void reverse(int a[],int n)
{int i,t;
  for(i=0;i<n/2;i++)
    {t=a[i];a[i]=a[n-1-i];a[n-1-i]=t;}
}
void main()
{int b[10]={1,2,3,4,5,6,7,8,9,10}; int i,s=0;
  reverse(b,8);
  for(i=6;i<10;i++) s+=b[i];
  printf("%d\n",s);
}

```

6、若有以下定义和语句，在程序中可通过指针数组 p，用\*(p[i]+j)等四种形式引用数组元素 s[i][j];另三种形式分别是： \*(&p[0][0]+6\*i+j)， \*(p+i)[j]和【 】。

```

int s[10][6], *p[10];
for(i=0;i<10;i++) p[i]=s[i];

```

7、下面程序运行后的输出结果是【 】。

```

#include <stdlib.h>
struct NODE
{int num; struct NODE *next; }
void main()
{struct NODE *p,*q,*r;
  p=(struct NODE *)malloc(sizeof(struct NODE));
  q=(struct NODE *)malloc(sizeof(struct NODE));
  r=(struct NODE *)malloc(sizeof(struct NODE));
  p->num=10;q->num=20;r->num=30;
  p->next=q;q->next=r;
  printf("%d\n",p->num+q->next->num);
}

```

---

特别提醒：答案必须写在答题纸上，若写在试卷或草稿纸上无效。



}

- 8、描述程序处理过程的工具称为【 】的工具，可以分为图形、表格和语言三类。
- 9、设计白盒测试方案的技术主要有逻辑覆盖和【 】。
- 10、根据软件规模可以估算出完成该项目所需的工作量，常用的估算模型为静态单变量模型、动态多变量模型和【 】模型。

## 二、判断题 ( 共 10 题 , 每题 1 分 , 共 10 分, 正确的标记 T,错误的标记 F )

- 1、sizeof()是 C 语言提供的标准库函数, 用于测试数据类型在系统中所占的字节数。
- 2、假定已有整型变量 x 的定义, 则(x+1)++是一个合法的表达式。
- 3、为了提高读写效率, 在进行读写操作后不应关闭文件以便下次再进行读写。
- 4、a 为二维数组, 则\*( &a[i]+j)与 a[i][j]的含义相同。
- 5、如果两个结构体变量有相同的类型, 则可以把其中一个结构体变量的值整体赋值给另一个结构体变量。
- 6、二重循环 for(i=0;i<5;i+=2) for(j=0;j<5;j++)的循环体的执行次数为 15。
- 7、静态外部变量只能在本文件使用, 其他文件不能使用该变量。
- 8、处于三级成熟度软件机构的过程能力可以概括为: 软件项目的策划和跟踪是稳定的, 已经为一个有纪律的管理过程提供了可重复以前成功实践的项目环境。
- 9、为了降低软件开发和维护的成本, 选用的高级语言应该具有良好的独立编译机制。
- 10、人机界面设计中关于信息显示的设计指南提出, 应该只显示与当前工作内容有关的信息。用户在获得有关系统特定功能的信息时, 不必看到与之无关的数据、菜单和图形。

## 三、单项选择题 ( 共 10 题 , 每题 2 分 , 共 20 分 )

- 1、有如下程序段:

```
int x1, x2;  
char y1, y2;  
scanf ("%d%c%d%c", &x1,&y1,&x2,&y2);
```

若要求 x1、x2、y1、y2 的值分别为 10、20、A、B, 正确的数据输入是 ( )。(注: □代表空格)

- A. 10A□20B
  - B. 10□A20B
  - C. 10 □A□20□ B
  - D. 10A20□B
- 2、下列描述中不正确的是 ( )。
- A. 字符型数组中可以存放字符串
  - B. 可以对字符型数组进行整体输入、输出
  - C. 可以对整型数组进行整体输入、输出

---

特别提醒: 答案必须写在答题纸上, 若写在试卷或草稿纸上无效。



D. 不能在赋值语句中通过赋值运算符"="对字符型数组进行整体赋值

3、设int a=12, 则执行完语句a+=a-=a\*a后, a的值是 ( )。

- A. 552      B. 264      C. 144      D. -264

4、设有如下枚举类型定义

```
enum language {Basic=3,Assembly=4,Ada=100,COBOL,Fortran};
```

枚举量Fortran的值为 ( )。

- A. 5      B. 6      C. 102      D. 103

5、下列程序执行后的输出结果是 ( )。

```
void f(int x,int y){
    int t;
    if(x<y)
        {t=x;x=y;y=t;}
}
```

```
void main() {
    int a=4,b=3,c=5;
    f(a,b);
    f(a,c);
    f(b,c);
    printf("%d,%d,%d\n",a,b,c);
}
```

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

6、下列程序执行后的输出结果是 ( )。

```
#include <stdio.h>
void main(){
    int i=1,j=2,k=3;
    if(i++==1&&(++j==3||k++==3))
        printf("%d %d %d",i,j,k); }
```

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

7、设有以下程序段, 则 ( )。

```
int x=0, s=0;
while(!x!=0) s+=++x; printf("%d",s);
```

- A. 运行程序段后输出 0      B. 运行程序段后输出 1  
C. 程序段中的表达式是非法的      D. 程序段执行无限次

8、软件测试的目的是 ( )。

- A. 避免软件中出现错误      B. 证明软件的正确性

---

特别提醒：答案必须写在答题纸上，若写在试卷或草稿纸上无效。

- C. 解决测试中发现的错误                      D. 发现软件中潜在的错误

9、下面 ( ) 说法是错误的。

- A. 测试应该尽早制定计划, 不断执行
- B. 软件错误具有聚集性, 对于存在错误的部分应该重点测试
- C. 软件测试是提高软件质量的决定性因素
- D. 测试用例需要定期评审和修改, 并且要不断增加新的测试用例

10、在攻关技术难题时, 最佳的软件开发团队组织模型是 ( )。

- A. 民主制程序员组                              B. 主程序员组
- C. 矩阵式程序员组                              D. 以上选项都不是

**四、程序分析 ( 共 3 题, 每题 10 分, 共 30 分。说明: 下划线部分表示程序的输入 )**

1、分析下面的程序并写出程序的执行结果。

```
#define maxn 110;
int val[maxn],vtop;
int op[maxn],otop;
void insert(int b)
{
    while(otop &&op[otop-1]==3)
    {
        b=!b;  --otop;
    }
    val[vtop++]=b;
}
```

```
void calc(void)
{
    int b=val[--vtop];
    int a=val[--vtop];
    int opr=op[--otop];
    int c=(a&b);
    if(opr==1)
        c=(a|b);
    insert(c);
}
```

```
int main(void) {
    int loop=0;
    char c;
    while((c=getchar())!=EOF)
```

---

特别提醒: 答案必须写在答题纸上, 若写在试卷或草稿纸上无效。



```

{
    vtop=otop=0;
    do
    {
        if(c=='(') {
            op[otop++]=0; }
        else if(c=='') {
            while(otop&&op[otop-1]!=0)
                calc();
            --otop;
            insert(val[--vtop]); }
        else if(c=='!') {
            op[otop++]=3; }
        else if(c=='&') {
            while(otop&&op[otop-1]>=2)
                calc();
            op[otop++]=2; }
        else if(c=='|') {
            while(otop&&op[otop-1]>=1)
                calc();
            op[otop++]=1; }
        else if(c=='V'||c=='F') {
            insert(c=='V'?1:0); }
    }
    while((c=getchar())!='\n'&&c!=EOF);
    while(otop)
        calc();

    printf("Expression %d: %c\n",++loop,(val[0]?'V':'F'));
}
return 0;
}

```

当输入为 (F&F|V|!V&!F&!(F|F&V)) 时，则程序的输出结果是：

2、分析下面的程序并写出程序的执行结果。

```

#include<stdio.h>
#define max 256
int n, m;
char DNA[10][10];

```

---

特别提醒：答案必须写在答题纸上，若写在试卷或草稿纸上无效。

```

char maxchar(int index) {
    int i,tmp=0, maxs;
    int W[max] = { 0 };
    char t;
    for (i = 0; i < m; i++) {
        t = DNA[i][index];      W[t]++;    }
    if (tmp < W['T']) {
        tmp = W['T'];      maxs = 'T';    }
    if (tmp < W['C']) {
        tmp = W['C'];      maxs = 'C';    }
    if (tmp < W['A']) {
        tmp = W['A'];      maxs = 'A';    }
    if (tmp < W['G']) {
        tmp = W['G'];      maxs = 'G';    }
    return maxs;
}

int main() {
    scanf("%d%d",&m,&n);
    int i, j;
    char str[10];
    for (i = 0; i < m; i++)    {
        scanf("%s",DNA[i]);
    }
    for (i = 0; i < n; i++) {
        printf("%c",maxchar(i));
    }
    return 0; }

```

输入:

5 8

TATGATAC

---

特别提醒：答案必须写在答题纸上，若写在试卷或草稿纸上无效。

TAAGCTAC

AAAGATCC

TGAGATAC

TAAGATGT

则程序的输出结果是:

3、分析下面的程序并写出程序的执行结果。

```
int GetSubString(char *strSource, char *strResult);
```

```
int main(){
```

```
    char strSource[]="ads3sl456789DF3456ld345AA", strResult[100];
```

```
    GetSubString(strSource, strResult);
```

```
    printf("strResult=%s\n",
```

```
    strResult, strSource);
```

```
    return 0;
```

```
}
```

```
int GetSubString(char *strSource, char *strResult){
```

```
    int iTmp=0, iHead=0, iMax=0;
```

```
    for(int Index=0, iLen=0; strSource[Index]; Index++)
```

```
    {
```

```
        if(strSource[Index] >= '0' && strSource[Index] <= '9' &&
```

```
            strSource[Index-1] > '0' && strSource[Index] == strSource[Index-1]+1) {
```

```
            iLen++;
```

```
        }
```

```
        else {
```

```
            if(iLen > iMax) {
```

```
                iMax = iLen;
```

```
                iHead = iTmp;
```

```
            }
```

```
            if(strSource[Index] >= '0' && strSource[Index] <= '9') {
```

```
                iTmp = Index;
```

```
                iLen = 1;
```

```
            }
```

```
        }
```

```
    }
```

```
    for(iTmp=0 ; iTmp < iMax; iTmp++)
```

```
        strResult[iTmp] = strSource[iHead++];
```

```
    strResult[iTmp]='\0';
```

```
    return iMax;
```

```
}
```

---

特别提醒：答案必须写在答题纸上，若写在试卷或草稿纸上无效。



五、程序填空 ( 共 3 题 , 每题 10 分 , 共 30 分 )

1、下面程序可求出以下方括号内的元素之积, 请填空。

|     |     |     |
|-----|-----|-----|
| [7] | 2   | 1   |
| 3   | [4] | 8   |
| 9   | 2   | [6] |

```
void main()
{ int x[3][3]={7,2,1,3,4,8,9,2,6};
  int s, *p;
  p=_____【1】_____;
  s=*p*_____【2】_____;
  printf(“product=%d”,s);
}
```

2、函数 factorsum(n) 用来计算正整数 n 除了本身外的因子和。例如: 如果 n 为 8, 它的非本身的因子是 1、2、4, 因此返回 7。请仔细阅读下面的程序并补充完整。

//八皇后递归解法

```
int p[1000][9];    //方式i 中第j行皇后的列位置为P[i][j]
int tmp[8];        //当前方式中第i行皇后的列位置为tmp[i]
int n=0;           //方式数初始化
bool col[8]={0},leftt[15]={0},rightt[15]={0}; // 所有列和左右对角线未被选中
void func(int r)   //从r行出发, 递归计算所有方案中8个皇后的位置
{
  if(r==8){
    for(int i=0;i<8;++i) //搜索所有行
      p[n][i]=tmp[i]; //记下当前方式中8个皇后的列位置
    n++;                // 方式数加一
    return;
  }
  for(int c=0;c<8;c++) //依次搜索r行的每一列
  {
```

---

特别提醒: 答案必须写在答题纸上, 若写在试卷或草稿纸上无效。

```

int ld=(c-r)+7;    //计算 (r,c) 的左右对角线序号
int rd=c+r;
if( 【1】 )//若第c列和左右对角线未选中，则选中c列和左右对角线
{
    col[c]=1,leftt[ld]=1,rightt[rd]=1;
    tmp[r]=c;    //(r,c) 放置皇后
    【2】    //递归下一行
    col[c]=0,leftt[ld]=0,rightt[rd]=0; //撤去第c列和左右对角线的选中标志
}
}
}

```

3、假设下标从 0 开始，0, 1, 2 .. m-1 共 m 个人，从 1 开始报数，报到 k 则此人从环出退出，第 i 个人出环的编号是多少？

设 f(m,k,i)函数递归方法计算 m 个人的环，报数为 k，第 i 个人出环的编号：

```

int fun(int m,int k,int i){
    if(i==1)
        return 【1】;
    else
        return 【2】;
}
int main(int argc, char* argv[]) {
    int sum,count,alive;
    printf("请输入要参与约瑟夫生存游戏的人数:(人数>0)\n");
    while(1) {
        scanf("%d",&sum);
        if(sum>0)
            break;
        printf("输入无效，请重新输入。 \n");
    };
    printf("请输入能报到的最大数字:(1<=数字)\n");
    while(1) {

```

---

特别提醒：答案必须写在答题纸上，若写在试卷或草稿纸上无效。



```

scanf("%d",&count);
if(count>=1)
    break;
printf("输入无效, 请重新输入.\n");
};
printf("请输入要求最后存活的人数:(0<=人数<=%d)\n",sum);
while(1)    {
    scanf("%d",&alive);
    if(alive>=0&&alive<=sum)
        break;
    printf("输入无效, 请重新输入.\n");
};
for(int i=1;i<=sum-alive;i++)
    printf("第%2d 次出环: %2d\n",i,fun(sum,count,i));
return 0;
}

```

## 六、程序设计 ( 共 2 题 , 第 1 题 10 分 , 第 2 题 15 分 , 共 25 分 )

1、给出一个区间的集合, 请你编写一个程序, 合并所有重叠的区间。

示例 1:

输入: [[1,3],[2,6],[8,10],[15,18]]

输出: [[1,6],[8,10],[15,18]]

解释: 区间 [1,3] 和 [2,6] 重叠, 将它们合并为 [1,6].

示例 2:

输入: [[1,4],[4,5]]

输出: [[1,5]]

解释: 区间 [1,4] 和 [4,5] 可被视为重叠区间。

2、在某城市里住着  $n$  个人, 任何两个认识的人不是朋友就是敌人, 而且满足: 1、我朋友的朋友是我的朋友; 2、我敌人的敌人是我的朋友; 所有是朋友的人组成一个团伙。告诉你关于这  $n$  个人的  $m$  条信息, 即某两个人是朋友, 或者某两个人是敌人, 请你编写一个程序, 计算出这个城市最多可能有多少个团伙?

输入:

第 1 行为  $n$  和  $m$ ,  $N$  小于 1000,  $M$  小于 5000; 以下  $m$  行, 每行为  $p \ x \ y$ ,  $p$  的值为 E 或 F,  $p$  为 F 时, 表示  $x$  和  $y$  是朋友,  $p$  为 E 时, 表示  $x$  和  $y$  是敌人。

输出:

一个整数, 表示这  $n$  个人最多可能有多少个团伙。

---

特别提醒: 答案必须写在答题纸上, 若写在试卷或草稿纸上无效。



样例输入：

6

4

E 1 4

F 3 5

F 4 6

E 1 2

样例输出：

3

提示：团伙构成集合{1},{2,4,6},{3,5}。

### 七、应用题 (共 2 题, 第 1 题 7 分, 第 2 题 8 分, 共 15 分)

1、用 Ashcroft\_Manna 技术可以将非结构化的程序转换为结构化程序, 图 1 是一个转换的例子。结合图示, 回答下面两个问题:

- (1) 从这个例子总结出 Ashcroft\_Manna 技术的一些基本方法。
- (2) 进一步简化图 1 (b) 给出的结构化设计。

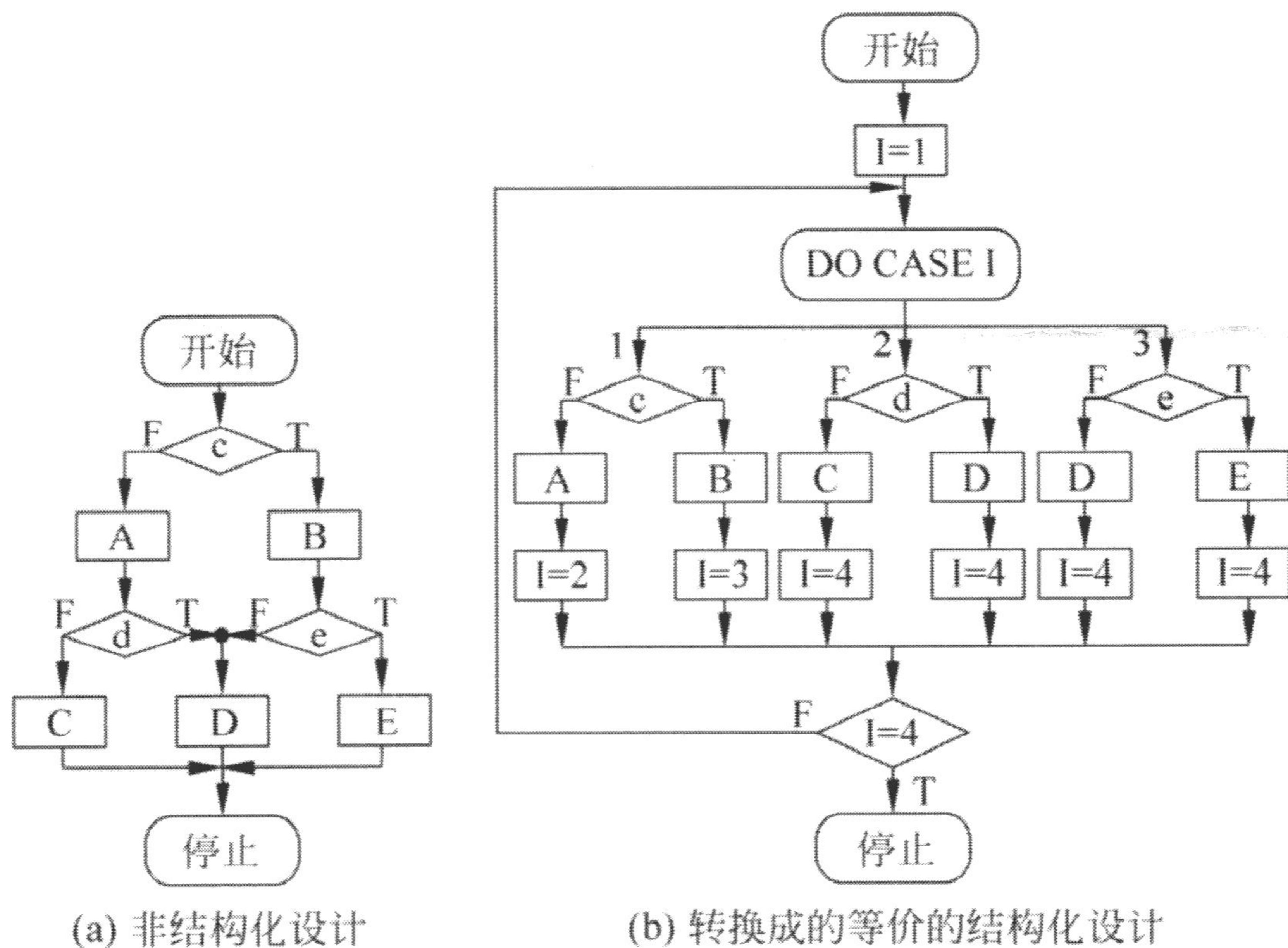


图 1 Ashcroft\_Manna 技术的例子

2、阅读下面代码, 画出该程序的流图, 并使用三种方法计算该流图的环形复杂度。

```
while (x<100) {
```

特别提醒：答案必须写在答题纸上, 若写在试卷或草稿纸上无效。



```
if (a[x]%2==0){
    parity = 0;
}else{
    parity = 1;
}
switch (parity) {
    case 0:
        println("a[" + i + "] is even");
    case 1:
        println("a[" + i + "] is odd");
    default:
        println("Unexpected error");
}
x++;
}
p = true;
```

---

特别提醒：答案必须写在答题纸上，若写在试卷或草稿纸上无效。