

希赛网, 专注于软考、PMP、通信考试的专业 IT 知识库和在线教育平台。希赛网在线题库, 提供历年考试真题、模拟试题、章节练习、知识点练习、错题本练习等在线做题服务, 更有能力评估报告, 让你告别盲目做题, 针对性地攻破自己的薄弱点, 更高效的备考。

希赛网官网: <http://www.educity.cn/>

希赛网软件水平考试网: <http://www.educity.cn/rk/>

希赛网在线题库: <http://www.educity.cn/tiku/>

2015 上半年程序员案例分析真题答案与解析: <http://www.educity.cn/tiku/tp20665.html>

## 2015 年上半年程序员考试下午真题

### (参考答案)

- 阅读以下说明和流程图, 填补流程图中的空缺, 将解答填入答题纸的对应栏内。

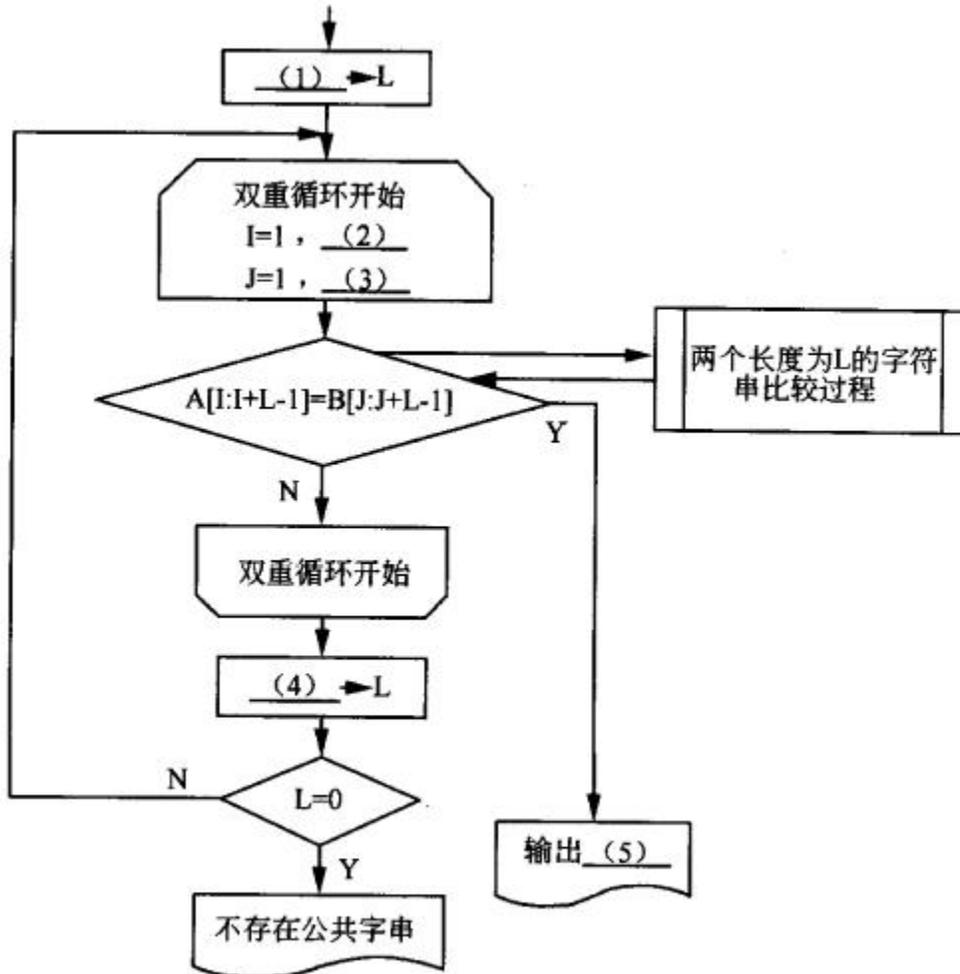
#### 【说明】

下面流程图的功能是: 在给定的两个字符串中查找最长的公共子串, 输出该公共子串的长度  $L$  及其在各字符串中的起始位置 ( $L=0$  时不存在公共子串)。例如, 字符串 "the light is not bright tonight" 与 "Tonight the light is not bright" 的最长公共子串为 "the light is not bright", 长度为 22, 起始位置分别为 2 和 10。

设  $A[1:M]$  表示由  $M$  个字符  $A[1], A[2], \dots, A[M]$  依次组成的字符串;  $B[1:N]$  表示由  $N$  个字符  $B[1], B[2], \dots, B[N]$  依次组成的字符串,  $M \geq N \geq 1$ 。

本流程图采用的算法是: 从最大可能的公共子串长度值开始逐步递减, 在  $A$ 、 $B$  字符串中查找是否存在长度为  $L$  的公共子串, 即在  $A$ 、 $B$  字符串中分别顺序取出长度为  $L$  的子串后, 调用过程判断两个长度为  $L$  的指定字符串是否完全相同 (该过程的流程略)。

#### 【流程图】



- 阅读以下说明和 C 函数，填补函数代码中的空缺，将解答填入答题纸的对应栏内。  
【说明 1】

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

函数 f(double eps) 的功能是:利用公式  $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$  计算并返回  $\pi$  的近似值。

**【C 函数 1】**

```
double f(double eps)
{
    double n = 1.0, s = 1.0, term = 1.0, pi = 0.0;
    while ( fabs(term) >= eps ){
        pi = pi + term;
        n = (1);          s = (2);
        term = s / n;
    }
    return pi*4;
}
```

**【说明 2】**

函数 fun(char \*str)的功能是:自左至右顺序取出非空字符串 str 中的数字字符, 形成一个十进制整数(最多 8 位)。例如, 若 str 中的字符串为 "iyt?67kp f3g8d5.j4ia2e3p12", 则函数返回值为 67385423。

**【C 函数 2】**

```
long fun(char *str)
{
    int i = 0;
    long num = 0;
    char *p = str;

    while ( i<8 && (3) ) {
        if ( *p >= '0' && *p <= '9' ) {
            num = (4) + *p - '0';
            ++i;
        }
        (5);
    }
    return num;
}
```

- 阅读以下说明和 C 代码, 填补代码中的空缺, 将解答填入答题纸的对应栏内。

**【说明】**

下面的程序代码根据某单位职工的月工资数据文件(名称为 Salary.dat, 文本文件),通过调用函数 GetIncomeTax 计算出每位职工每月需缴纳的个人所得税额并以文件(名称为 IncomeTax.dat, 文本文件)方式保存。

例如, 有 4 个职工工资数据的 Salary.dat 内容如下, 其中第一列为工号(整数), 第 2 列为月工资(实数)。

```
1030001 6200.00
1030002 5800.00
2010001 8500.00
2010010 8000.00
```

相应地, 计算所得 IncomeTax.dat 的内容如下所示, 其中第 3 列为个人所得税额:

```
1030001 6200.00 47.20
1030002 5800.00 35.94
2010001 8500.00 233.50
2010010 8000.00 193.00
```

针对工资薪金收入的个人所得税计算公式为:

个人所得税额=应纳税所得额×税率-速算扣除数

其中, 应纳税所得额=月工资-三险一金一起征点

税率和速算扣除数分别与不同的应纳税所得额对应, 如表 3-1 所示。

表 3-1

级数	全月应纳税所得额 X (元)	税率 (%)	速算扣除数
1	$0 < X \leq 1500$	3	0
2	$1500 < X \leq 4500$	10	105
3	$4500 < X \leq 9000$	20	555
4	$9000 < X \leq 35000$	25	1005
5	$35000 < X \leq 55000$	30	2755
6	$55000 < X \leq 80000$	35	5505
7	$X > 80000$	45	13505

设三险一金为月工资的 19%, 起征点为 3500 元。

例如, 某人月工资为 5800 元, 按规定 19% 缴纳三险一金, 那么:

其应纳税所得额  $X = 5800 - 5800 \times 19\% - 3500 = 1198$  元, 对应税率和速算扣除数分别为 3% 和 0 元, 因此, 其个人所得税额为  $1198 \times 3\% - 0 = 35.94$  元。

**【C代码】**

```

#include <stdio.h>
#define BASE 3500           //起征点
#define RATE 0.19         //三险一金比例
____(1)____;               //声明函数 GetIncomeTax
int main()
{
    int id;
    double salary;
    FILE *fin,*fout;

    fin = fopen("Salary.dat","r");
    if (____(2)____) return 0;
    fout = fopen("IncomeTax.dat","w");
    if (____(3)____) return 0;

    while (!feof(fin)) {
        if (fscanf(fin,"%d%lf",____(4)____)!=2) break;
        fprintf(fout,"%d\t%.2lf\t%.2lf\n",id,salary,____(5)____);
    }
    fclose(fin);
    fclose(fout);
    return 0;
}

double GetIncomeTax(double salary)
{
    double yns_sd;

    yns_sd = ____ (6) ____ - BASE;    /*计算应纳税所得额*/
    if (yns_sd<=0)    return 0.0;
    else if (yns_sd<=1500)    return yns_sd*0.03;
    else if (yns_sd<=4500)    return yns_sd*0.1 - 105;
    else if (yns_sd<=9000)    return yns_sd*0.2 - 555;
    else if (yns_sd<=35000)    return yns_sd*0.25 - 1005;
    else if (yns_sd<=55000)    return yns_sd*0.3 - 2755;
    else if (yns_sd<=80000)    return yns_sd*0.35 - 5505;
    return yns_sd*0.45 - 13505;
}

```

- 阅读以下说明和 C 函数，填补代码中的空缺，将解答填入答题纸的对应栏内。

**【说明】**

函数 Combine(LinkList La, LinkList Lb)的功能是:将元素呈递减排列的两个带头结点单链表合并为元素值呈递增(或非递减)方式排列的单链表,并返回合并所得单链表的头指针。例如,元素递减排列的单链表 La 和 Lb 如图 4-1 所示,合并所得的单链表如图 4-2 所示。

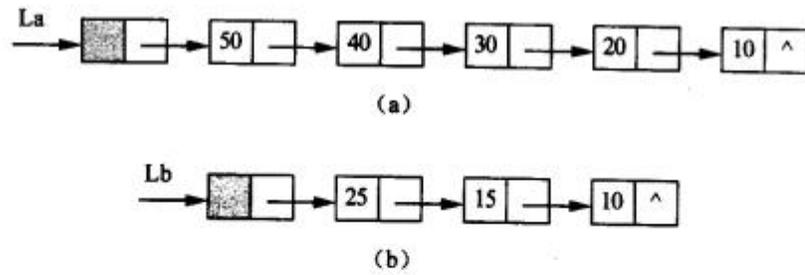


图 4-1 合并前的两个链表示意图

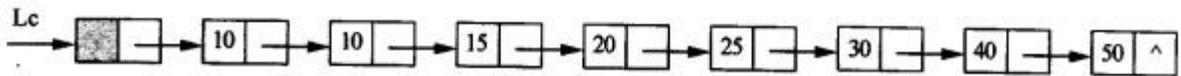


图 4-2 合并后所得链表示意图

设链表结点类型定义如下:

```
typedef struct Node{
    int data;
    struct Node *next;
}Node,*LinkList;
```

**【C 函数】**

```

LinkList Combine(LinkList La, LinkList Lb)
{ //La 和 Lb 为含头结点且元素呈递减排列的单链表的头指针
  //函数返回值是将 La 和 Lb 合并所得单链表的头指针
  //且合并所得链表的元素值呈递增 (或非递减) 方式排列

  (1) Lc, tp, pa, pb;; //Lc 为结果链表的头指针, 其他为临时指针

  if (!La) return NULL;
  pa = La->next; //pa 指向 La 链表的第一个元素结点

  if (!Lb) return NULL;
  pb = Lb->next; //pb 指向 Lb 链表的第一个元素结点

  Lc = La; //取 La 链表的头结点为合并所得链表的头结点
  Lc->next = NULL;

  while ( (2) ){ //pa 和 pb 所指结点均存在 (即两个链表都没有到达表尾)
    //令 tp 指向 pa 和 pb 所指结点中的较大者
    if (pa->data > pb->data){
      tp = pa; pa = pa->next;
    }
    else{
      tp = pb; pb = pb->next;
    }
    (3) = Lc->next; //tp 所指结点插入 Lc 链表的头结点之后
    Lc->next = (4) ;
  }

  tp = (pa)? pa : pb; //设置 tp 为剩余结点所形成链表的头指针

  //将剩余的结点合并入结果链表中, pa 作为临时指针使用
  while (tp) {
    pa = tp->next;
    tp->next = Lc->next;
    Lc->next = tp;
    (5) ;
  }

  return Lc;
}

```

- 阅读下列说明和 C++ 代码, 填补代码中的空缺, 将解答填入答题纸的对应栏内。

**【说明】**

设计 RGB 方式表示颜色的调色板, 进行绘图, 其类图如图 5-1 所示。该程序的 C++ 代码附后。

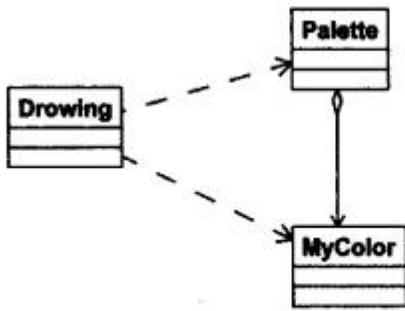


图 5-1 类图

希赛网在线题库

**【C++代码】**

```
#include <iostream>
#include <stdlib.h>
#include <ctime>
using namespace std;
class MyColor{
private:
    int red;    int green;    int blue;
public:
    MyColor(){red = 0; green = 0; blue = 0; }
    ~MyColor(){}
    MyColor(int red, int green, int blue) {
        this->red = red; this->green = green; this->blue = blue;}
//其他方法略
    void print() {
        cout<<"Red: " << red << "\tGreen: " << green << "\tBlue " << blue
        << endl;
    }
};
class Palette {
private:
    int number; MyColor** palette;
public:
    Palette(){ number = 256; palette = (MyColor**)malloc
        (sizeof(MyColor*)*number); }
    ~Palette(){
        for (int i = 0; i < number; i++) { delete palette[i]; }
        (1);
    }
    Palette(MyColor** pale, int number) {
        (2) = number;
        palette = (MyColor**)malloc(sizeof(MyColor*)*number);
        memcpy(palette, pale, sizeof(pale)*number);
    }
//其他方法略
    void print() {
        for (int i = 0; i < number; i++) {
            cout << i << " : " ;
            palette[i]->print();
        }
    }
};
class Drawing{
public:
    (3) int COLORNUMBER = 16;
public:
    ~Drawing() { }
    void draw() {
        Palette* palette;
        int red, green, blue;
        MyColor* color[COLORNUMBER];
        srand((unsigned)time(0));
        for (int i = 0; i < COLORNUMBER; i++) {
```

- 阅读以下说明和 Java 代码，填补代码中的空缺，将解答填入答题纸的对应栏内。

**【说明】**

设计 RGB 方式表示颜色的调色板，进行绘图。其类图如图 6-1 所示。该程序的 Java 代码附后。

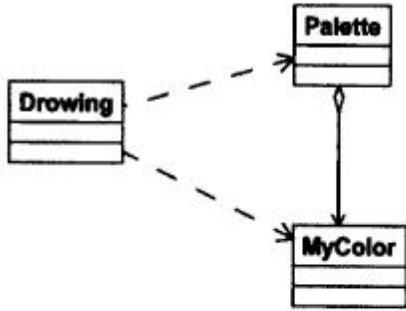


图 6-1 类图

**【Java 代码】**

```
//颜色类
class MyColor {
    private int red, green, blue;

    public MyColor() {
        red = 0; green = 0; blue = 0;
    }
    public MyColor(int red, int green, int blue) {
        this.red = red;
        this.green = green;
        this.blue = blue;
    }
}

//其他方法略
public String toString() {
    return "Red: " + red + "\tGreen: " + green + "\tBlue " + blue;
}

}

//调色板类
class Palette {
    public int number;           //颜色数
    private (1) palette;       //颜色表

    public Palette() {
        number = 256;
        palette = new MyColor[number];
    }
    public Palette(MyColor[] palette, int number) {
        (2) = number;
        (3) = palette;
    }
}

//其他方法略

public String toString() {
    String str = "";
    for (int i = 0; i < number; i++) {
        str += i + " : " + palette[i] + "\n";
    }
    return str;
}

}

//绘图类
class Drawing {
    public (4) int COLORNUMBER = 16;

    public static void main(String[] args)    {
        Palette palette;
        int red, green, blue;
        MyColor[] color = new MyColor[COLORNUMBER];
        for (int i = 0; i < COLORNUMBER; i++) {
            red = (int)(Math.random() * 256);
            green = (int)(Math.random() * 256);
            blue = (int)(Math.random() * 256);
            color[i] = (5)(red, green, blue);
        }
        palette = new Palette(color, COLORNUMBER);
        System.out.println(palette);
    }
}
```

希赛网在线题库