

希赛网, 专注于软考、PMP、通信、建造师、教资等考试的专业 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, 备考更高效。

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/>

2018 年上半年程序员考试下午真题答案与解析:

<http://www.educity.cn/tiku/tp41603.html>

## 2018 年上半年程序员考试下午真题

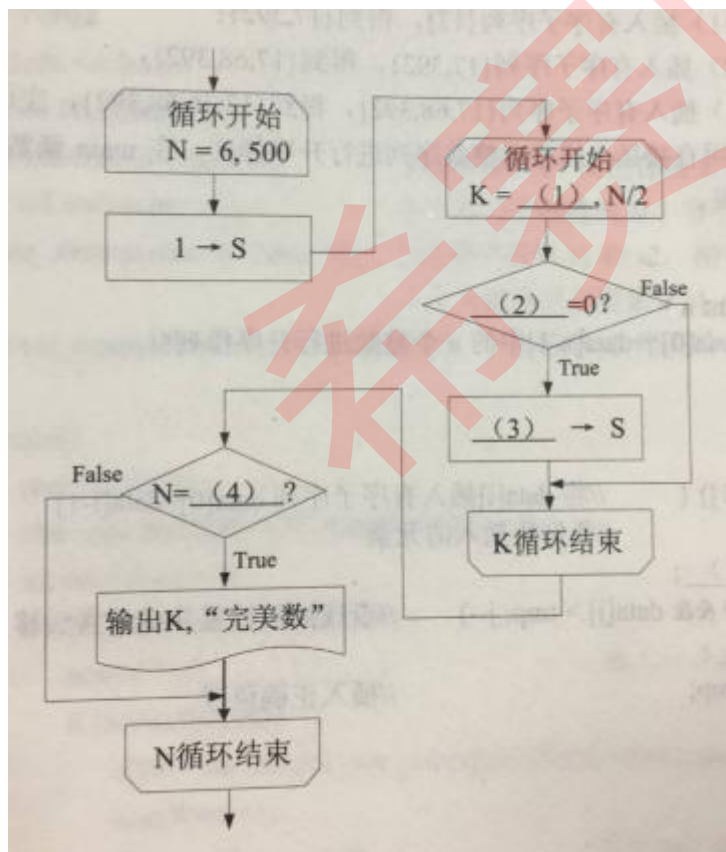
● 阅读以下说明和流程图, 填写流程图中的空缺, 将解答填入答题纸的对应栏内。

### 【说明】

如果一个自然数  $N$  恰好等于它所有不同的真因子 (即  $N$  的约数以及 1, 但不包括  $N$ ) 之和  $S$ , 则称该数为“完美数”。例如  $6=1+2+3$ ,  $28=1+2+4+7+14$ , 所以 6 和 28 都是完美数。显然, 6 是第 1 个 (即最小的) 完美数。

下面流程图的功能是求 500 以内所有的完美数。

### 【流程图】



循环开始框中要注明: 循环变量=初始值, 终值 [, 步长], 步长为 1 时可以缺省。

如果某自然数小于其所有真因子之和 (例如  $24 < 1+2+3+4+6+8+12$ ), 则称该自然数为亏数: 如果某自然数大于其所有真因子之和 (例如  $8 > 1+2+4$ ), 则称该自然数为赢数 如果某自然数等于从 1 开始的若干个连续自然数之和 (例如  $10 = 1+2+3+4$ ) 则称该自然

数为三角形数数。据此定义, 自然数 496 是 ( )。

供选择答案:

- (1) A. 亏数      B. 赢数      C. 完美数, 非三角形数      D. 完美数和三角形数

● 阅读以下说明和 C 代码, 填写程序中的空 (1) ~ (5), 将解答写入答题纸的对应栏内

**【说明】**

直接插入排序是一种简单的排序方法, 具体做法是: 在插入第  $i$  个关键码时,  $k_1, k_2, \dots, k_{i-1}$  已经排好序, 这时将关键码  $k_i$  依次与关键码  $k_{i-1}, k_{i-2}, \dots$ , 进行比较, 找到  $k_i$  应该插入的位置时停下来, 将插入位置及其后的关键码依次向后移动, 然后插入  $k_i$ 。

例如, 对 {17, 392, 68, 36} 按升序作直接插入排序时, 过程如下:

第 1 次: 将 392 ( $i=1$ ) 插入有序子序列 {17}, 得到 {17, 392};

第 2 次: 将 68 ( $i=2$ ) 插入有序子序列 {17, 392}, 得到 {17, 68, 392};

第 3 次: 将 36 ( $i=3$ ) 插入有序子序列 {17, 68, 392}, 得到 {17, 36, 68, 392}, 完成排序。

下面函数 insert Sort 用直接插入排序对整数序列进行升序排列, 在 main 函数中调用 insert Sort 并输出排序结果。

**【C 代码】**

```
void insert Sort(int data[], int n)
/*体用直接插入排序法将 data [0] ~ data[n-1] 中的 n 个整数进行升序排列*/
{ int i, j; int tmp;
  for(i=1; i < n; i++) if (data[i] < data [i-1]) { //将 data[i] 插
    入有序子序列 data [0] ~data [i-1]
      tmp=data[i]; //备份待插入
    的元素
      data[i]= (1)
      for(j=i-2; j>=0&&data[ j]>tmp; j---) //查找插入位
    置并将元素后移
      (2) ;
      (3) =tmp; //插入正确位
    置
  } /*if*/
} /*for*/
} /*insert Sort*/

int main ()
{ int*bp, *ep;
  int n, arr[ ]={17, 392, 68, 36, 291, 776, 843, 255}
  n=size of(arr)/size of(int);
  insert Sort(arr, n);
  bp=( 4 ); ep=arr+n;
  for(; bp < ep; bp++) printf("%d\t", (5) );
  return 0;
}
```

● 阅读以下 C 代码, 回答问题 1 至问题 3, 将解答填入答题纸的对应栏内。

**【C 代码 1】**

```
float adjust Salary(int service_year,int age,float salary){
    if(service_year<=5){
        if(age>30)
            salary*=1.2;
    }
    else
        salary*=2.0;
    return salary;
}
```

**【C 代码 2】**

```
void foo(int coin){
    switch (coin){
        case 1: printf("Cent\n");
        case 5: printf("Nickel\n");break;
        case 10: printf("Dime\n");
        case 25: printf("Quarter\n");
    }
}
```

**【C 代码 3】**

```
int get Sum(int n)
{
    Int t,i=0,sum=0;
    while(i<n){
        scanf( "%d",&t);
        if(t<0)
            continue;
        sum+=t
        i++;
    }
    return sum;
}
```

**【问题 1】 (3 分)**

对于【C 代码 1】, 写出下面的函数调用后 x1、x2 和 x3 的值。

x1=adjust Salary(1,25,9000.0);

x2=adjust Salary(8,32,7000.0);

x3=adjust Salary(5,41,5000.0);

**【问题 2】 (6 分)**

(1) 写出函数调用为 foo(1) 的输出结果;

- (2) 写出函数调用为 foo (5) 的输出结果  
 (3) 写出函数调用为 foo (10) 的输出结果;  
 (4) 写出函数调用为 foo (25) 的输出结果。

【问题 3) (6 分)

- (1) 简要说明【C 代码 3】中的函数 get Sum () 的功能;  
 (2) 程序员小王将函数 get Sum 改写后得到下面的函数 get Sum\_v2 (即用 for 语句来描述循环)。请问, get Sum\_v2 的功能是否与【C 代码 3】的 get Sum 完全相同, 若不同请说明原因。

```
int get Sum_v2(int n)
{
    int t, i=0, sum=0;
    for(i=0; i<n; i++)
        scanf("%d", &t);
        if(t<0)
            continue;
        sum+=t;
    }
    return sum;
}
```

● 试题四 (共 15 分)

阅读以下说明和 C 代码, 填写代码中的空 (1) ~ (6), 将解答写入答题纸的对应栏内。

【说明】

下面的 C 代码在输入的 100 个英文单词中找出最小单词和最大单词。约定每个单词是仅由英文字母构成的字符串, 且都不超过 20 个字符。单词词的大小按照字典序定义。例如, 单词 “entry” 大于 “enter”、“art” 小于 “article”、“an” 等于 “An”。

【C 代码】

```
#include <stdio.h>
#define NUMBER 100
int is Valid(const char*s1); //若字符串 s1 仅包含英文字母则返回 1,
否则返回 0
char to Lower(char ch); //将大写字母转换为小写字母
int usr_strcpy (char*s1, char*s2); //比较字符串 s1 和 s2, 相等时返回 0,
//s1 大
则返回正整数, s1 小则返回负整数
void usr_strcpy (char*s1, const char*s2); //字符串 s2 拷贝给 s1
```

```
int main__ (4) __
{
    char word [32];
    char max Word[32]="", min Word[32] ="";
    int num Word=0;
    while(num Word<NUMBER) {
        scanf("%s", (1)); //输入一个单词存入 word
        if(is Valid(word)) {
            if(0==num Word) {usr_strcpy(min
```

```

Word, word):usr_strcpy(max Word, word);}
        num Word++;
        if( ____ (2) ____ >0) //调用
usr_strcpy 比较单词
        usr_strcpy (max Word, word); //用 max Word 记下
最大单词
        else
        if( ____ (3) ____ <0) //调用 usr_strcpy 比较
单词
        usr_strcpy (min Word, word); //用 min Word 记下
最小单词
    }
}
printf("max Word=%s min Word=%s\n",max Word,min Word);
return 0;
}
int is Valid(const char*s)
{
    for(;*s;s++)
        if(!(*s>=' a' &&*s<=' z' )&&!(*s>=' A' &&*s<=' Z' ))
            return 0;
    return 1;
}

Char to Lower(char ch)
{//若 ch 为大写字母则返回其小写形式, 否则直接返回原字符
    if(ch>=' A' &&ch<=' Z' )
        ch=(4)+' a' ;
    return ch;
}

Int usr_strcpy(char*s1, char*s2)
{ //按字典序比较两个英文单词, 若 s1 表示的单词大, 则返回正整数,
//若 s1 表示的单词小, 则返回负整数; 否则返回 0

for(; ____ (5) ____;){
    if(to Lower(*s1)==to Lower(*s2)) {s1++, s2++;}
    else
        break;
}
return(to Lower(*s1)-to Lower(*s2));
}

```

```
void usr_strcpy(char*s1, const char*s2)
{
    //将 s2 表示的字符串复制给 s1
    for(;;)
        *s1++=*s2++;
    *s1=' \0' ;
}
```

● 试题五 (共 15 分)

阅读以下说明和 Java 程序, 填写程序中的空 (1) ~ (6), 将解答写入答题纸的对应栏内

【说明】

以下 Java 代码实现一个简单绘图工具, 绘制不同形状以及不同颜色的图形。部分接口、类及其关系如图 5-1 所示。

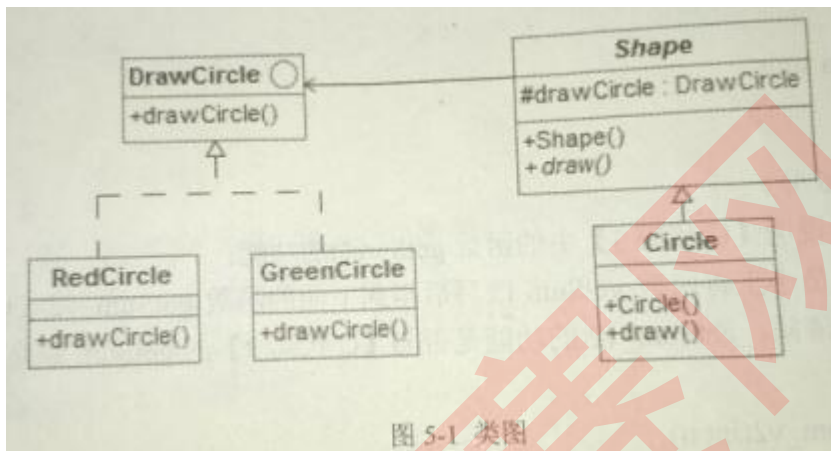


图 5-1 类图

【Java 代码】

```
interface Draw Circle { //绘制圆形
    public (1) ;
}

class Red Circle implements Red Circle { //绘制红色圆形
    public void draw Circle(int radius, int x, int y) {
        System.out.print
        In("Drawing Circle[red, radius:" + radius + ", X:" + x + ", y:" + y + "]");
    }
}

class Green Circle implements Draw Circle { //绘制绿色圆圆形
    public void draw Circle(int radius, int x, int y) {
        System.out.print
        In("Drawing Circle[green, radius:" + radius + ", X:" + x + ", y:" + y + "]");
    }
}

abstract class Shape { //形状
    protected (2) ;
}
```

```

public Shape(Draw Circle draw Circle){
    this.draw Circle= draw Circle;
}
    public abstract void draw__(5)__;
}

class Circle extends Shape { //圆形
    private int x,y,radius;

    public Circle(int x,int y,int radius,Draw Circle draw Circle){
        _____(3)_____ ;
        this.x=x;
        this.y=y;
        this.radius=radius;
    }

    public void draw( ){
        draw Circle. _____(4)_____ ;
    }
}

public class Draw Circle Main{
    public static void main(String[]args){
        Shape red Circle=new Circle (100,100,10, __(5) ) ; //绘制红色圆
形
        Shape green Circle=new Circle (2000,200,10, __(6) ) ; //绘制绿色圆
形

        red Circle.draw__(6)__;
        green Circle.draw__(7)__;
    }
}

```

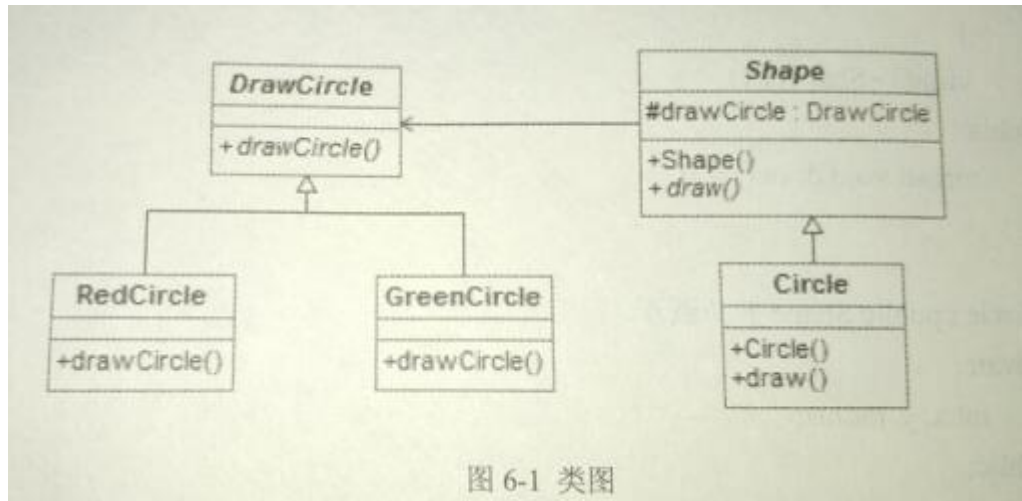


#### 试题六 (共 15 分)

阅读下列说明和 C++ 代码, 填写程序中的空 (1) ~ (6), 将解答写入答题纸的对应栏内。

#### 【说明】

以下 C++ 代码实现一个简单绘图工具, 绘制不同形状以及不同颜色的图形。部分类及其关系如图 6-1 所示。



**【C++代码】**

```

#include <iostream>
#include <string>
using namespace std;
class Draw Circle { //绘制圆形, 抽象类
public:
    _____(1); //定义参数为 int radius, int x, int y
    virtual ~Draw Circle__(6)__( )
};

class Red Circle: public Draw Circle { //绘制红色圆形
public:
    void draw Circle(int radius, int x, int y) {
        cout<<"Drawing Circle[red, radius:"<<radius;
        cout<<" , X:"<<X<<" , y:"<<y<<"]"<<endl;
    }
};

class Green Circle: public Draw Circle { //绘制绿色圆形
public:
    void draw Circle(int radius, int x, int y) {
        cout <<"Drawing Circle[green, radius:"<<radius;
        cout<<" , X:"<<X<<" , y:"<<y<<"]"<<endl;
    }
};

class Shape { //形状, 抽象类
protected:
    _____(2)_____ ;
public:
    Shape(Draw Circle*draw Circle) {
    
```



```

        this->draw Circle=draw Circle;
    }
    virtual~shape__(7)__{}
public:
    virtual void draw__(8)__=0;
};

class Circle: public Shape { //圆形
private:
    int x,y,radius;
public:
    Circle(int x,int y,int radius,Draw Circle*draw
Circle)    (3)  {
        this->x=x;
        this->y=y;
        this->radius=radius;
    }
Public:
    void draw__(9){
        draw Circle ->    (4)  ;
    }
};

int main__(10)__
{
    Shape * red Circle = new Circle
(100,100,10,    (5)  ); //绘制红色圆形
    Shape * green Circle = new Circle
(100,100,10,    (6)  ); //绘制绿色圆形
    red Circle ->draw__(11__);
    green Circle ->draw__(12__);
    return 0;
}

```