

希赛网 (www.educity.cn) 专注于在线教育服务 18 年, 拥有海量学员见证。是软考行业的开拓者与推动机构, 自成希赛体系的培训系统。负责软考教材编排与评审, 出版了 80% 以上辅导教材。全职自有师资直播+录播双保障教学保障, 高精度做题和知识系统, 助力软考学员一次通关。

希赛软考: <http://www.educity.cn/rk>

希赛题库: <http://www.educity.cn/tiku>

2019 上半年程序员下午真题答案与解析:

<https://www.educity.cn/tiku/tp340372.html>

2019 上半年程序员下午真题

1、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。

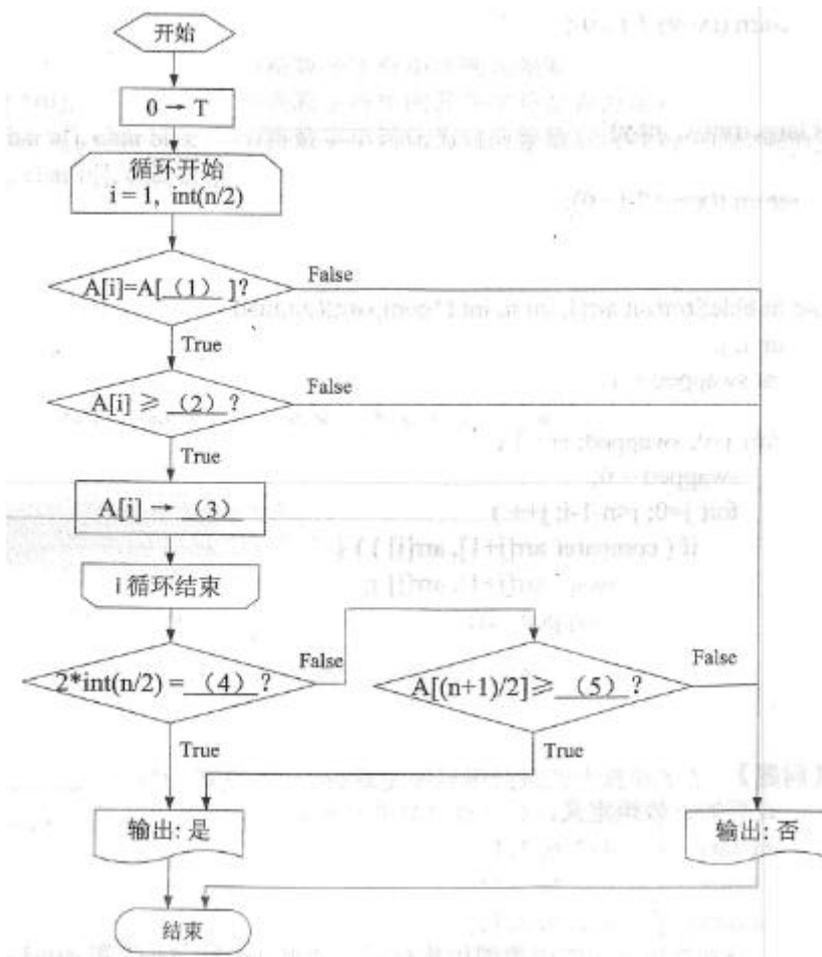
【说明】

如果 n 位数 ($n \geq 2$) 是回文数 (从左到右读与从右到左读所得结果一致), 且前半部分的数字递增 (非减)、后半部分的数字将递减 (非增), 则称该数为拱形回文数。例如, 12235753221 就是一个拱形回文数。显然, 拱形回文数中不含数字 0。

下面的流程图用于判断给定的 n 位数 (各位数字依次存放在数组的各个元素 $A[i]$ 中, $i=1, 2, \dots, n$) 是不是拱形回文数。流程图中, 变量 T 动态地存放当前位之前一位的数字。当 n 是奇数时, 还需要特别注意中间一位数字的处理。

问题内容:

【流程图】



注 1: “循环开始”框内给出循环控制变量的初值、终值和增值（默认为 1），格式为: 循环控制变量=初值, 终值 [, 增值]

注 2: 函数 $\text{int}(x)$ 为取 x 的整数部分, 即不超过 x 的最大整数。

2、阅读以下说明和 C 代码, 回答问题, 将解答写入答题纸的对应栏内。

【说明】

函数 `bubbleSort(int arr[], int n, int (*compare)(int,int))` 的功能是根据调用时传递的比较函数 `compare` 对数组 `arr` 的前 `n` 个元素进行排序。

【C 代码】

```

#define swap(a, b) { a = a ^ b; b = a ^ b; a = a ^ b; } // 交换 a 与 b 的值
int less(int x, int y)
{
    return ((x < y) ? 1 : 0);
}
int larger(int x, int y)
{
    return ((x > y) ? 1 : 0);
}
  
```

```

void bubbleSort(int arr[ ], int n, int (*compare)(int,int))
{
    int i,j;
    int swapped = 1;
    for( i=0; swapped; i++) {
        swapped = 0;
        for(j=0; j<n-1-i; j++)
            if ( compare( arr[j+1], arr[j])) {
                swap( arr[j+1], arr[j]);
                swapped =1;
            }
    }
}

```

问题内容:

【问题】

设有如下数组定义:

```
int data1[ ] = {4, 2, 6, 3, 1};
```

```
int data2[ ] = {4, 2, 6, 3, 1};
```

```
int data3[ ] = {4, 2, 6, 3, 1};
```

请分别给出下面的函数调用执行后, 数组 data1、data2 和 data3 各自的元素序列。

- (1) bubbleSort(data1, 5, less);
- (2) bubbleSort(data2, 5, larger);
- (3) bubbleSort(data3, 3, larger);

3、阅读以下说明和 C 代码, 填写程序中的空缺, 将解答写入答题纸的对应栏内。

【说明】

某市根据每天早上 5 点测得的雾霾指数 (pm2.5 值) 决定是否对车辆进行限行。

规则 如下:

- (1) 限行时间为周内。(即周一到周五), 周六周日不限行;
- (2) 根据车牌号的尾号 (最后 1 位数字) 设置限行车辆 (车牌号由英文字母和十进制数字构成, 长度为 6 位, 至少包含 1 位数字);
- (3) 雾霾指数低于 200 时, 不限行;
- (4) 雾霾指数在区间 [200, 400) 时, 周内每天限行两个尾号的汽车: 周一限行 1 和 6, 周二限行 2 和 7, 周三限行 3 和 8, 周四限行 4 和 9, 周五限行 5 和 0, 即尾号除以 5 的 余数相同者在同一天限行;
- (5) 雾霾指数大于等于 400 时, 周内每天限行五个尾号的汽车: 周一、周三和周五限行 1,3,5,7,9, 周二和周四限行 0,2,4,6,8, 即尾号除以 2 的余数相同者在同一天限行;

下面程序运行时, 输入雾霾指数、星期(数字 1 表示星期一, 数字 2 表示星期二, ..., 数字 7 表示星期日) 和车牌号, 输出该车牌号是否限行的信息。

问题内容:

【C 代码】

```
#include <stdio.h>
#define PM25_L1 200
#define PM25_L2 400
typedef enum {YES,NO} MARKTAG;
int isDigit (char ch)
{//判断 ch 是否为十进制数字字符，是则返回 1，否则返回 0
    return (ch>='0' && ch<='9') ;
}
void prt_msg(char *msg, MARKTAG flag)
{
    if (flag == YES)
        printf("%s : traffic restrictions\n", msg);
    else
        printf("%s : free\n", msg);
}

int isMatched(int weekday, int t, int d) //判断是否符合限行规则，是则返回 1，否则返回 0
{ return (weekday%d == t%d); }
void proc(int pm25, int weekday, char *licence)
{
    int i,lastd;
    if (weekday == 6 || weekday == 7 || (1) )
        prt_msg(licence, NO);
    else {
        for( i=5; i>=0; i-- )
            if (isDigit(licence[i])) {
                lastd= (2) ;//获取车牌号的尾号
                break;
            }
        if(pm25>= PM25_L2) { //限行 5 个尾号的汽车
            if (isMatched( (3) ))
                prt_msg(licence, YES);
            else
                prt_msg(licence, NO);
        }

        else { //限行 2 个尾号的汽车
            if (isMatched( (4) ))
                prt_msg(licence, YES);
            else
                prt_msg(licence, NO);
        }
    }
}
```

```

    }
}
int main()
{
    int weekday=0, pm25=0;
    char licence[7];
    scanf("%d %d %s", &pm25, &weekday, licence);
    //输入数据的有效性检测略，下面假设输入数据有效、正确
    proc(    (5)    );
    return 0;
}

```

4、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】

下面程序运行时，对输入的表达式进行计算并输出计算结果。设表达式由两个整数和一个运算符（+或-）构成，整数和运算符之间以空格分隔，运算符可以出现在两个整数之前、之间或之后，整数不超过 4 位，输入的两个整数和运算符都用字符串表示。

例如，输入分别为“25+7”、“+25 7”、“25 7+”时，输出均为“25 + 7 = 32”。

问题内容：

【C 代码】

```

#include<stdio.h>
int str2int (char *s); //将数字字符串转换为整数
int isOperator (char *str); //判断字符串的开头字符是否为运算符
void cal (char op, char a[], charb[]); //将数字串转化为对应整数后进行 op 所要求的计算
void solve (char a[], char b[], char c[]);

int main ()
{
    char a[10],b[10], c[10];
    scanf ("%s%s%s", a, b, c);
    //输入数据的有效性检测略，下面假设输入数据有效、正确
    Solve(a,b,c);
    Return0;
}
int str2int(char *s)
{

```

```

int val = 0;
while (*s) {
    val = (1) + (*s - '0'); //将数字字符串
转换为十进制整数
    (2) ; //令
    字符指针指向下一个数字字符
}
return val;
}

int isOperator(char *str)
{
    return (*str == '+' || *str == '-');
}

void cal( char op, char a[], char b[])
{
    switch(op) {
        case '+':
            printf(" %s + %s = %d", a, b, str2int(a)+str2int(b));
            break;
        case '-':
            printf(" %s - %s = %d ", a,b,str2int(a)-str2int(b));
            break;
    }
}

void solve (char a[], char b[], char c[])
{//解析输入的 3 个字符串，输出表达式及计算结果
    if (isOperator(a)) { //运算符在两个整数之前
        cal( (3) );
    }
    else if(isOperator(b)) { //运算符在两个整数之间
        cal( (4) );
    }
    else { //运算符在两个整数之后
        cal( (5) )
    }
}

```

5、阅读以下说明和 Java 代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

【说明】

现如今线下支付系统可以使用现金（Cash）、移动支付、银行卡（Card）（信用卡（CreditCard）和储蓄卡（DebitCard））等多种支付方式（PaymentMethod）对物品（Item）账单（Bill）进行支付。图 5-1 是某支付系统的简略类图。

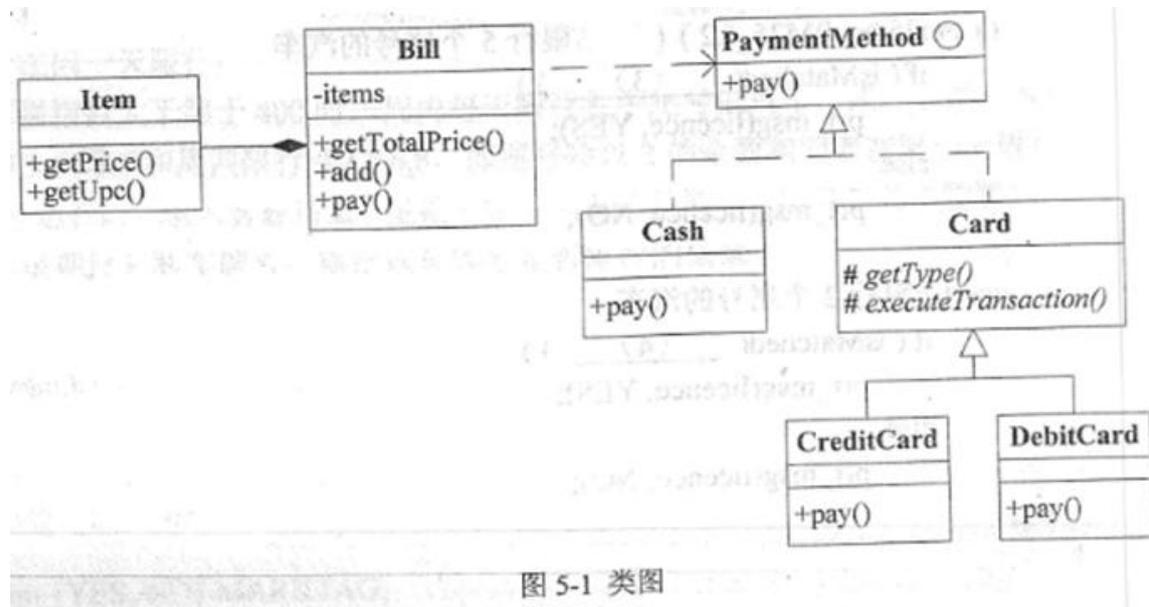


图 5-1 类图

问题内容：

【Java 代码】

```

import java.util. ArrayList;
import java.util. List;
interface PaymentMethod {
    public      (1)      ;
}
// Cash、DebitCard 和 Item 实现略，Item 中 getPrice()获取当前物品对象的价格
abstract class Card      (2)      {
    private final String name, num;
    public Card(String name, String num) {this.name = name; this.num = num; }
    @Oveirde
    public String toString      ( )      {
        return String.format("%s card[name = %s, num = %s]", this.getType (),
name, num);
    }
    @Override
    public void pay(int cents) {

        System.out.println("Payed " + cents + " cents using " + toString( ));
        this.executeTransaction(cents);
    }
    protected abstract String getType( );
    protected abstract void executeTransaction(int cents);
}
    
```

```

}
class CreditCard      (3)      {
    public CreditCard(String name, String num) {      (4)      ;}
    @Override
    protected String getType( ) { return "CREDIT"; }
    @Override
    protected void executeTransaction(int cents) {
        System.out.println(cents + " paid using Credit Card. ");
    }
}

class Bill { //包含所有购买商品的账单
    private List<Item> items = new ArrayList<>();
    public void add(Item item) { items.add(item); }
    public int getTotalPrice( ) /*计算所有 item 的总价格，代码略*/
    public void pay(PaymentMethod paymentMethod) { //用指定的支付方式完成
支付
        (5)      (getTotalPrice( ));
    }
}

public class PaymentSystem {
    public void pay( ) {
        Bill bill = new Bill( );
        Item item1 = new Item("1234",10); Item item2 = new Item( "5678",40);
        bill.add(item1); bill.add(item2); //将物品添加到账单中
        bill.pay(new CreditCard("LI SI", "98765432101")); //信用卡支付
    }

    public static void main(String[ ] args) {
        (6)      = new PaymentSystem( );
        payment.pay( );
    }
}
}

```

6、阅读下列说明和 C++代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

【说明】

现如今线下支付系统可以使用现金 (Cash)、移动支付、银行卡 (Card) (信用卡 (CreditCard) 和储蓄卡 (DebitCard)) 等多种支付方式 (PaymentMethod) 对物品 (Item) 账单 (Bill) 进行支付。图 6-1 是某支付系统的简略类图。

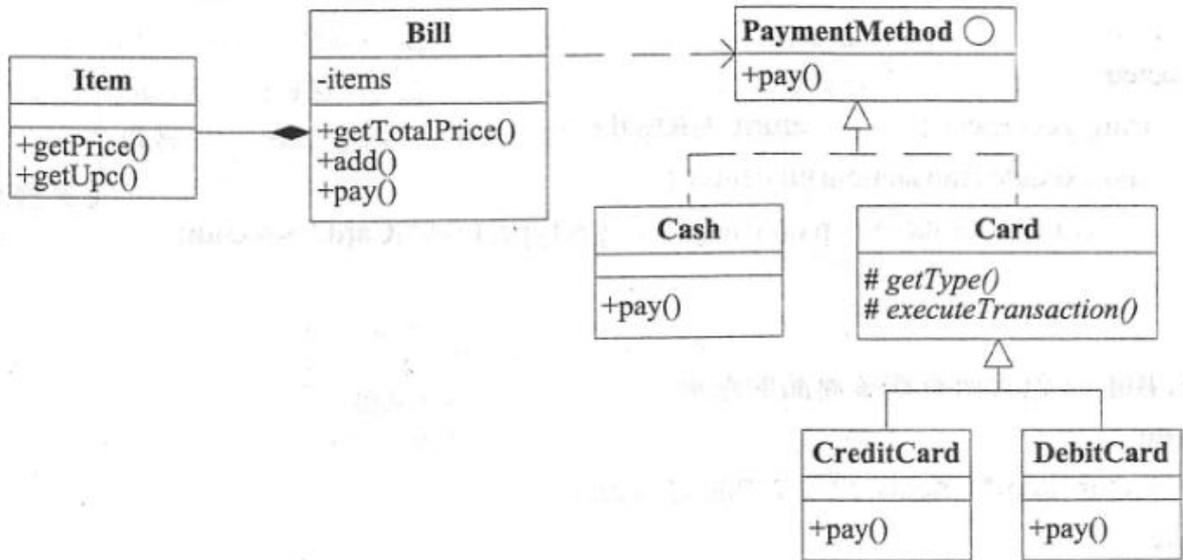


图 6-1 类图

问题内容:

【 C++代码 】

```

#include <iostream>
#include <vector>
#include <string>
using namespace std;
class PaymentMethod {
public: virtual void pay(int cents)=0;
};
// Cash、DebitCard 和 Item 实现略，Item 中 getPrice()获取当前物品对象的价格
class Card : public PaymentMethod {
private: string name, num;
public:
    Card(string name, string num) {this->name = name; this->num = num; }
    string toString() {
        return this->getType() + " card[name = " + name + ", num = " + num +
    "]" ; }
    void pay(int cents) {
        cout<< "Payed " <<cents << " cents using " <<toString() <<endl;
        this->executeTransaction(cents);
    }
protected:
    virtual string getType()=0;
    virtual void executeTransaction(int cents)=0;
};
  
```

```

class CreditCard    (2)    {
public:
    CreditCard(stringname, stringnum)    (3)    {
    }
protected:
    string getType() { return " CREDIT " ; }
    void executeTransaction(int cents) {
        cout<<cents << " paid using " <<getType()<< " Card." << endl;
    }
};

class Bill { //包含所有购买商品的账单
private:
    vector< Item*> items; //包含物品的 vector
public:
    void add(Item* item){ items.push_back(item); }
    int getTotalPrice() { /*计算所有 item 的总价格，代码略*/ }
    void pay(PaymentMethod* paymentMethod) { //用指定的支付方式完成支付
        (4)    (getTotalPrice());
    }
};

class PaymentSystem{
public:
    void pay() {
        Bill* bill = new Bill();
        Item* item1= new Item("1234", 10); Item* item2 = new
Item("5678",40);
        bill->add(item1); bill->add(item2); //将物品添加到账单中
        (5)    (new CreditCard("LI SI", "98765432101")); //信用卡支付
    }
};

int main() {
    (6)    = new PaymentSystem();
    payment->pay();
    return 0;
}

```