

希赛网, 专注于**软考**、**PMP**、**通信**、**建造师**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/>

2017 年下半年程序员考试下午真题答案与解析:

<http://www.educity.cn/tiku/tp31384.html>

2017 年下半年程序员考试下午真题

试题一(共 15 分)

阅读以下说明和流程岛, 填补流程图中的空缺, 将解答填入答题纸的对应栏内。

【说明】

对于大于 1 的正整数 n , $(x+1)^n$ 可展开为

$$C_n^0 x^n + C_n^1 x^{n-1} + C_n^2 x^{n-2} + \dots + C_n^{n-1} x^1 + C_n^n x^0$$

下面流程图的作用是计算 $(x+1)^n$ 展开后的各项系数 C_n^i ($i=0,1,\dots,n$) 并一次存放在数组 $A[0\dots n]$ 中, 方法是依次计算 $k=2, 3, \dots, n$ 时 $(x+1)^k$ 的展开系数并存入数组 A , 在此过程中, 对任一确定的 k , 利用关系式 $C_k^i = C_{k-1}^i + C_{k-1}^{i-1}$, 按照 i 递减的顺序逐步计算并将结果存储在数组 A 中。其中, C_2^0 和 C_k^k 都为 1, 因此可直接设置 $A[0]$ 、 $A[k]$ 的值为 1。

例如, 计算 $(x+1)^3$ 的过程如下:

先计算 $(x+1)^2$ (即 $k=2$) 的各项系数, 然后计算 $(x+1)^3$ (即 $k=3$) 的各项系数。

$k=2$ 时, 需要计算 C_2^0 , C_2^1 和 C_2^2 并存入 $A[0]$, $A[1]$ 和 $A[2]$, 其中 $A[0]$ 和 $A[2]$ 的值已有,

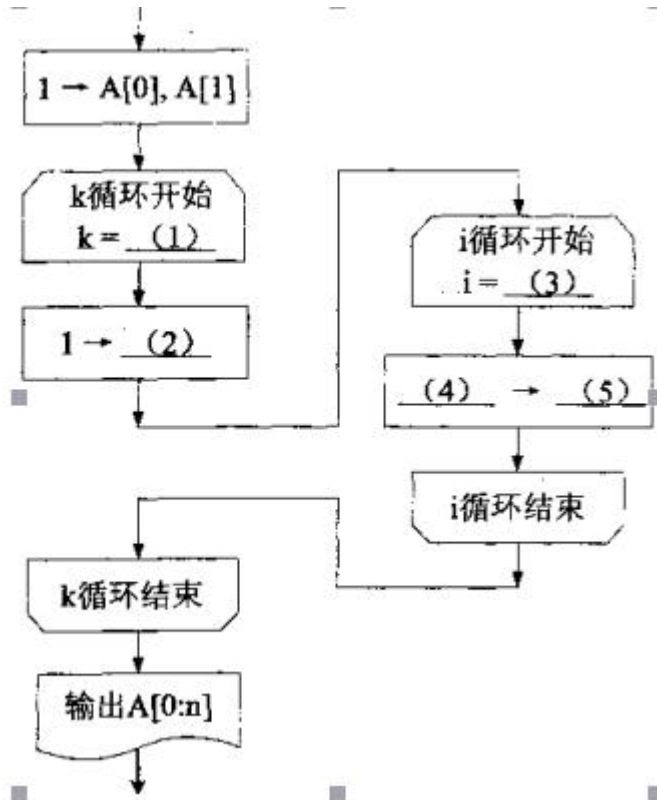
因此将 C_1^1 (即 $A[1]$) 和 C_1^0 (即 $A[0]$) 相加得到 C_2^1

的值并存入 $A[1]$

$k=3$ 时, 需要计算 C_3^0 , C_3^1 和 C_3^2 和 C_3^3 , 先计算出 C_3^2 (由 $C_2^2 + C_2^1$) 得到并存入

$A[2]$, 再计算 C_3^1 (由 $C_2^1 + C_2^0$) 得到并存入 $A[1]$ 。

【流程图】



注:循环开始框内应给出循环控制变量的初值和终值,默认递增值为 1。
格式为: 循环控制变量=初值, 终值, 递增值。

试题二 (共 15 分)

阅读以下说明和代码, 填补代码中的空缺, 将解答填入答题纸的对应栏内。

【说明】

对 n 个元素进行简单选择排序的基本方法是: 第一趟从第 1 个元素开始, 在 n 个元素中选出最小者, 将其交换至第一个位置, 第二趟从第 2 个元素开始, 在剩下的 $n-1$ 个元素中选出最小者, 叫将其交换至第二个位置, 依此类推, 第 i 趟从 $n-i+1$ 个元素中选出最小元素, 将其交换至第 i 个位置, 通过 $n-1$ 趟选择最终得到非递减排序的有序序列。

【代码】

```

#include <stdio.h>
Void selectSort(int data[ ],intn)
//对 data[0]~data[n-1]中的 n 个整数按非递减有序的方式进行排列
{
inti,j,k;
Int temp
for(i=0;i<n-1;i++){
For(k=1,j=i+1;____;____)
//k 表示 data[1]~data[n-1]中最小元素的下标
If(data[j]<data[k])____;
If(k!=i){

```

```
//将本趟找出的最小元素与 data[i]交换
Temp=data[i];_____ ;data[k]=temp;
Int main()
Int arr[ ]={79.85.93.65.44.70.100.57};
Inti,m,
m=sizeof(arr)/sizeof(int);_____
For(_____;i<m;i++)
Prittf(“%d/t”,arr[i]);
Printf(“/n”);
Return 0
//计算数组元素的个数, 用 m 表示
//调用 seletSort 对数组 arr 进行非递减排序
//按非递减顺序输出所有的数组元素
```

试题三(共 15分)

阅读以下代码和问题, 回答问题 1 至问题 33 将解答填入答题纸的对应栏内。

【代码1】

```
typedefenum {ABCD} EnumType;
EnumType f(int yr)
if(0=yr%400)
return A
else if (!yr%4)
if(0!=yr%100)
return B
else
return C
return D
```

【问题1】(4分)

对于代码 1, 写出下面的函数调用后 x1、x2、x3 和 x4 的值。

```
X1=f(1997);
x2 = f(2000);
x3 = f(2100);
x4 = f(2020);
```

【问题2】

```
#include<stdio.h>
int main()
int score;
scanf(“%d”, &score);
switch (score)
case 5: printf(“Excellent!\n”);
case 4: printf(“Good! \n”); break;
case 3: printf(“Average! \n”);
```

```
case2:  
case1:  
case 0: printf("Poor!\n");  
default: printf("Oops,Erro \n");
```

【问题2】

- (1) 写出代码 2 进行时输入为 3 的输出结果
- (2) 写出代码 2 进行时输入为 5 的输出结果

【代码3】

```
#include<stdio.h>  
int main()  
int i,j,k;  
for(i=0; <2; i++)  
for(j=0; j<3;j++)  
for( k=0; k<2;k==)  
if(i!=j7&j!=k)  
printf("%d o/od o/od\n", i,j,k);
```

【问题3】

写出代码 3 运行后的输出结果

试题四(共 15 分)

阅读以下说明、 C 函数和问题, 回答问题 1 和问题 2 将解答填入答题纸的对应栏内。

【说明】

当数组中的元素已经排列有序时, 可以采用折半查找(二分查找)法查找一个元素。下面的函数 biSearch(int r[], int low, int high, int key)用非递归方式在数组 r 中进行二分查找, 函数 biSearch_rec(int r[], int low, int high, int key)采用递归方式在数组 r 中进行二分查找, 函数的返回值都为所找到元素的下标;若找不到,则返回-1。

【函数 1】

```
int biSearch(int r[], int low, int high, int key)  
//r[low..high] 中的元素按非递减顺序排列  
//用二分查找法在数组 r 中查找与 key 相同的元素  
//若找到则返回该元素在数组 r 的下标,否则返回-1
```

```
int mid;  
while(____){  
mid = (low+high)/2 ;  
if (key ==r[mid])  
Return mid;  
else if (key<r[mid])_____  
else  
_____  
}/*while*/
```

```
return-1;
}/*biSearch*/
```

【问题1】 (12分)

请填写 C 函数 1 和 C 函中 2 中的空缺, 将解答填入答题纸的对应栏。

【问题2】 (3分)

若有序数组中有 n 个元素, 采用二分查找法查找一个元素时, 最多与一个数组元素进行比较, 即可确定查找结果。

(7) 备选答案

- A. $\lceil \log_2(n+1) \rceil$
- B. $\lfloor n/2 \rfloor$
- C. n-1
- D. n

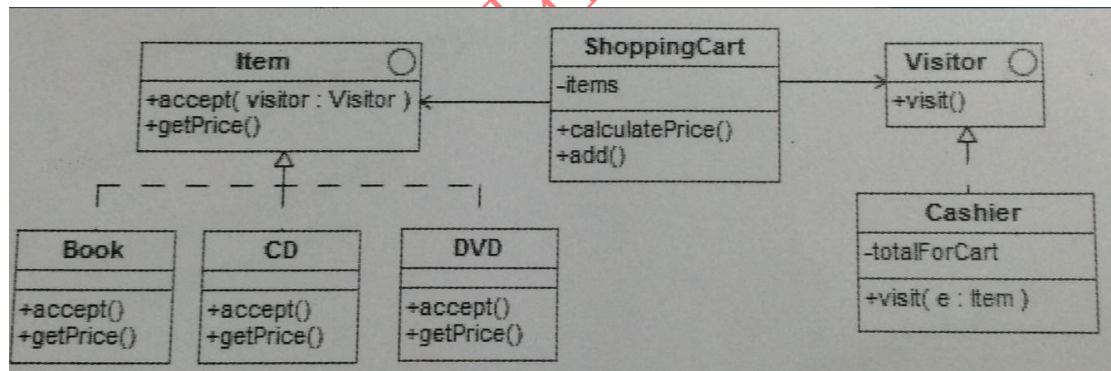
从下 4 列的 2 道试题(试题五至试题六)中任选 1 道解答。请在答题纸上的指定位置处将所选择试题的题号框涂黑。若多涂或者未涂题号框, 则对题号最小的一道试题进行评分。

试题五 (共 15 分)

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】

以下 Java 代码实现一个超市简单销售系统中的部分功能, 顾客选择图书等物件 (Item) 加入购物车 (ShoppingCart), 到收银台 (Cashier) 对每个购物车中的物统计其价格进行结账。设计如图 5-1 所示类图。



【Java 代码】

```
interface Item{
public void accpt(Visitor visitor);
public double getPrice();
classBook
private double price;
classBook(double price)
public void accpt(Visitor visitor)
public double getPrice()
return price;
```

//其它物品类略

```
interface Visitor
public void visit(Book book);
//其他物品的 visit 方法
class Cashier
public double totalForCart;
//访问 Boo 类型对象的价格并累加
//假设 Book 类型的物品价格超过 10 元打 8 折
if(book.getPrice()<10.0)
totalForCart+=book.getPrice()
else
totalForCart+=book.getPrice()*0.8;
//其他 visit 方法和折扣策略类似。此处略
public double getTotal()
return totalForCart

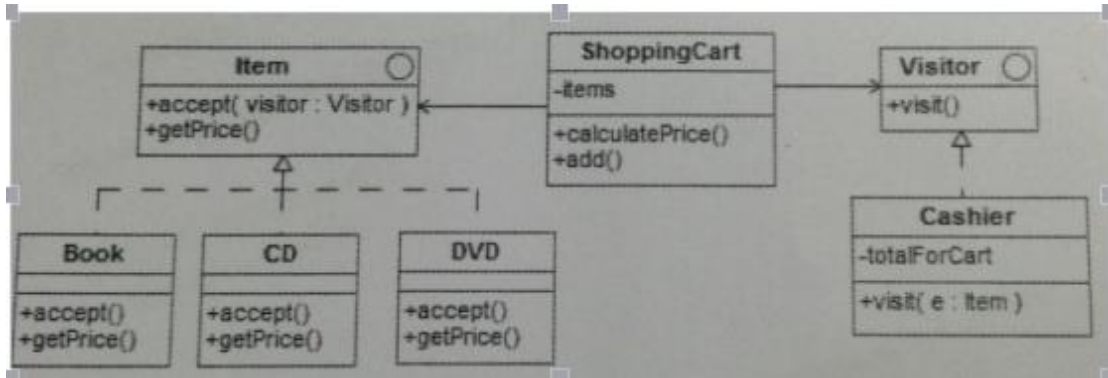
Class ShoppingCart
//normal shopping cart stuff
private java.util.ArrayList<Item>items=new java.util.ArrayList<>();
public double calculatePrice()
Cashier visitor=new Cashier()
For(Item item:items)
double total=visitor.getTatal()
return total
public void add(Item e)
this.items.add(e)
```

试题六(共四分)

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】

以下 C++代码实现一个超市简单销售系统中的部分功能,顾客选择图书等物品 (Items) 加入购物车(SgoppingCart),到收银台(Cashier) 对每个购物车中的物品统计其价格进行结账. 设计如图 6-1 所示类图。



【C++代码】

Using namespace std;

classbook

class Visist

public:

Virtual void visit(Book* book)=0;

//其它物品的 visit 方法

Class Item

Public:virtual void acctp(Visit*visit)=0;

Virtual double getPrice()=0

class BOOK

public:

Book (double price){//访问本元素——;}

Void accept (Visitor*visitor){_}

double getPrice()

{return price;}

class Cashier_{

private

doule totalForCart

Public:

//访问 Book 类型队形的价格并累加__

//假设 Book 类型的物品价格超过 10 元打 8 折

if(book->getPrice()>10.0)

{totalForCart+=book->getprice();

}else

totalForCart+=book->getPrice()*0.8;

}

//其他 visit 方法和折扣策略类似, 此忽略

double getTotal(){

return tatalForCart;

class ShoppingCart{

```
private:
vector<Item*>itenms;
public:
double calculatePrice(){
Csahier*visitor=new Cashier();
For(int i=0;i<items.size();i++){_;}
Double total=visitor->getTatal();
Return tatal;
Void add(Item*e)
{items.push_back (e) ;}
```

希赛网在线题库