**全国计算机技术与软件专业技术资格（水平）考试**

**2017年下半年 程序员 下午试卷**

（考试时间 14:00～16:30 共 150 分钟）

|  |
| --- |
| **请按下述要求正确填写答题纸** |

1.在答题纸的指定位置填写你所在的省、自治区、直辖市、计划单列市的名称。

2.在答题纸的指定位置填写准考证号、出生年月日和姓名。

3.答题纸上除填写上述内容外只能写解答。

4.本试卷共6道题，试题一至试题四是必答题，试题五至试题六选答 1 道。每

题 15 分，满分 75 分。

5.解答时字迹务必清楚，字迹不清时，将不评分。

6.仿照下面例题，将解答写在答题纸的对应栏内。

**例题**

2017 年下半年全国计算机技术与软件专业技术资格（水平）考试日期是（1）

月（2）日。

因为正确的解答是“11 月 4 日”，故在答题纸的对应栏内写上“11”和“4”

（参看下表）。

|  |  |
| --- | --- |
| 例题 | 解答栏 |
| （1） | 11 |
| （2） | 4 |

**试题一**

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】**

http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-1.png

对于大于1的正整数n，(x+1)n可展开为

下面流程图的作用是计算(x+1)n展开后的各项系数http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-2.png（i=0,1,....,n）并依次存放在数组A[0...n]中。方法是依次计算k=2，3，..，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http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-3.png，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，http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-4.png和http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-5.png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：

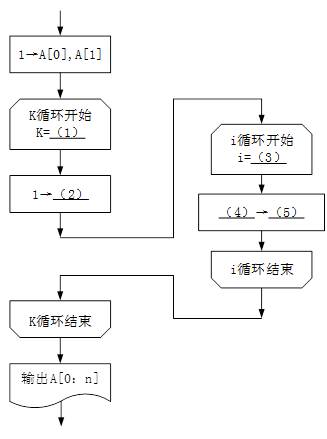
先计算 (x+1)2 ( 即k=2) 的各项系数，然后计算(x+1)3(即k=3)的各项系数。

K=2时，需要计算http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-6.png，http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-7.png和http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-8.png，并存入A[0] ，A[1]和A[2]，其中A[0]和A[1]的值已有，因此将http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-9.png（即 A[1]）和http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-10.png即 (A[0])相加得到http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-17.png的值并存入A[1]。

k=3 时，需要计算http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-11.png，http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-12.png和http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-13.png和http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-14.png，先计算出http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-131.png (由http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-15.png)得到并存入A[2]，再计算http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-121.png (由http://www.rkpass.cn:8080/ruankao_work_version_0103/userfile/image/cxy2017-x-x-1d-16.png得到)并存入A[1]。

**【问题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**

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，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。

**【问题1】**

**【代码】**

#include <stdio.h>

void selectSort(int data[ ],int n)

//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

{

int i,j,k;

int temp;

for(i=0;i<n-1;i++){

for(k=i,j=i+1;（1）;（2）) //k表示data[i]~data[n-1]中最小元素的下标

if(data[j]<data[k]) （3）

if(k!=i) {

//将本趟找出的最小元素与data[i]交换

temp=data[i];（4）;

data[k]=temp;

}

}

}

int main()

{

int arr[ ]={79,85,93,65,44,70,100,57};

int i,m;

m=sizeof(arr)/sizeof(int); //计算数组元素的个数，用m表示

（5）; //调用selectSort对数组arr进行非递减排序

for(（6）;i <m;i++) //按非递减顺序输出所有的数组元素

printf(“%d\t”,arr[i]);

printf(“\n”);

return 0;

}

**试题三**

阅读以下代码和问题，回答问题1至问题3，将解答填入答题纸的对应栏内。

**【代码 1】**

typedef enum {A，B，C，D} 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;

}

【代码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");

case 2:

case 1:

case 0: printf("Poor!\n");

default: printf("Oops,Error\n");

}

return 0;

}

【代码3】

#include<stdio.h>

int main()

{

int i,j,k;

for(i=0; i<2; i++)

for(j=0; j<3;j++)

for( k=0; k<2;k++) {

if(i!=j&&j!=k)

printf("%d %d %d\n", i,j,k);

}

ruturn 0;

}

**【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】（5分）**

（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。

**【问题3】（6分）**

写出代码3运行后的输出结果。

**试题四**

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。

**【C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(（1）) {

mid = (low+high)/2 ;

if (key ==r[mid])

return mid;

else if (key<r[mid])

（2）;

else

（3）;

}/\*while\*/

return -1;

}/\*biSearch\*/

**【C 函数 2】**

int biSearch\_rec(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;

if(（4）) {

mid = (low+high)/2 ;

if (key ==r[mid])

return mid;

else if (key<r[mid])

return biSearch\_rec(（5）,key);

else

return biSearch\_rec(（6）,key);

}/\*if\*/

return -1;

}/\*biSearch\_rec\*/

**【问题1】 (12分）**

请填充C函数1和C函数2中的空缺，将解答填入答题纸的对应栏内。

**【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（ ）个数组元素进行比较，即可确定查找结果。

（7）备选答案：

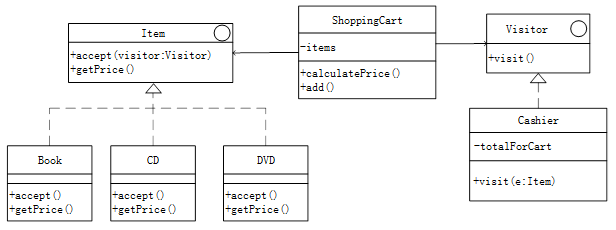
A.[log2(n+1)] B.[n/2] C.n-1 D.n

**试题五**

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品统计其价格进行结账。设计如图5-1所示类图。



**图5-1类图**

**【问题1】**

**【Java代码】**

interface Item{

public void accept(Visitor visitor);

public double getPrice();

}

class Book （1）{

private double price;

public Book(double price){（2）;}

public void accept(Visitor visitor){ //访问本元素

（3）;

}

public double getPrice() {

return price;

}

}

//其它物品类略

interface Visitor {

public void visit(Book book);

//其它物品的visit方法

}

class Cashier（4）{

private double totalForCart;

//访问Book类型对象的价格并累加

（5）{

//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;

}

//其它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) {

（6）;

}

double total=visitor.getTotal();

return total;

}

public void add(Item e) {

this.items.add(e);

}

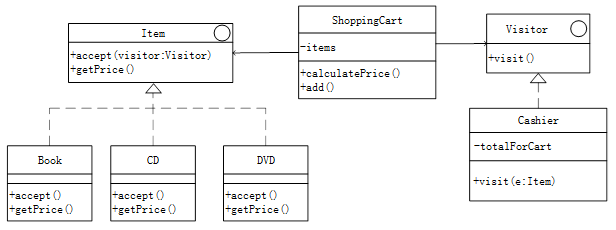
}

**试题六**

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（Item）加入购物车（ShoppingCart），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。



**图6-1 类图**

**【问题1】**

**【C++代码】**

using namespace std;

class Book;

class Visitor {

public:

virtual void visit(Book\* book)=0;

//其它物品的visit方法

};

class Item {

public:virtual void accept(Visitor\* visitor)=0;

virtual double getPrice()=0;

};

class Book （1）{

private: double price;

public:

Book (double price){ //访问本元素

（2）;

}

void accept(Visitor\* visitor) {

（3）;

}

double getPrice() { return price; }

};

class Cashier（4）{

private;

double totalForCart;

public:

//访问Book类型对象的价格并累加

（5）{

//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;

}

//其它visit方法和折扣策略类似，此处略

double getTotal() {

return totalForCart;

}

};

class ShoppingCart {

private:

vector<item\*>items;

public:

double calculatePrice() {

Cashier\* visitor=new Cashier();

for(int i=0;i <items.size();i++)

（6）;

}

double total=visitor->getTotal();

return total;

}

void add(Item\*e) {

items.push\_back（e）;

}

};