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/

2016 下半年程序员案例分析真题答案与解析：http://www.educity.cn/tiku/tp19679.html

2016 年下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。

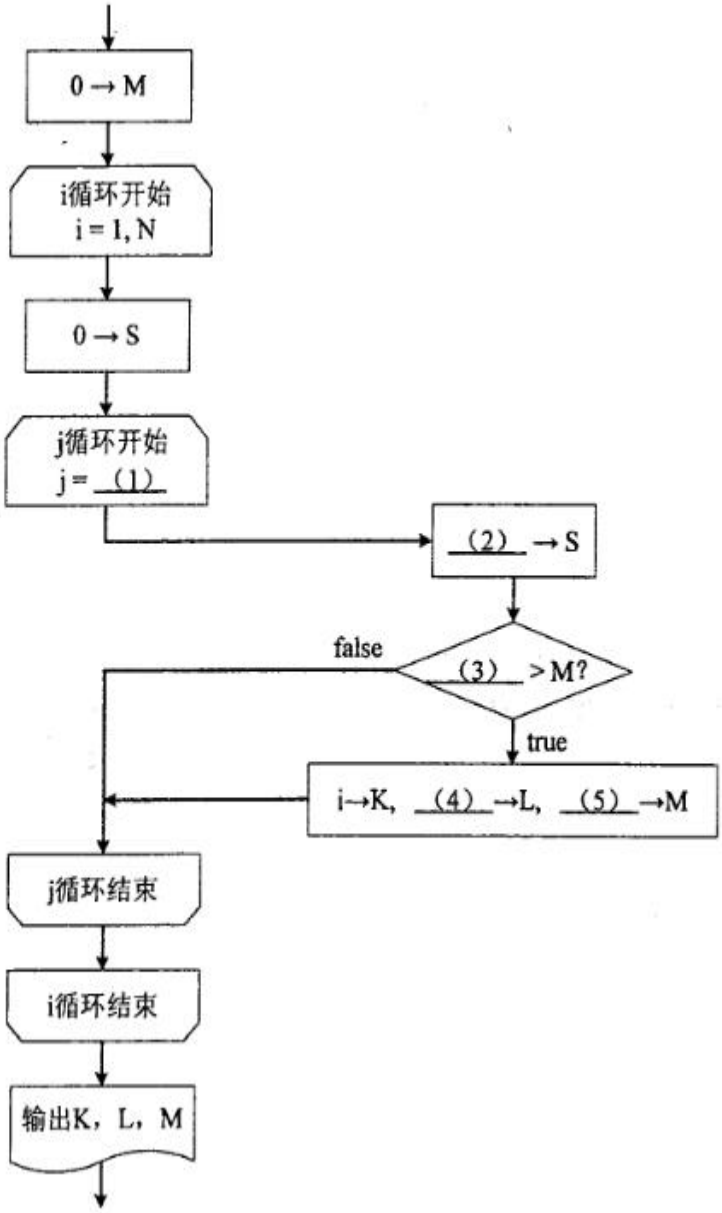
【说明】

设有整数数组 A[1：N]（N>1），其元素有正有负。下面的流程图在该数组中寻找连续排列的若干个元素，使其和达到最大值，并输出其起始下标 K、元素个数 L 以及最大的和值 M。

例如，若数组元素依次为 3，-6，2，4，-2，3，-1，则输出 K=3，L=4,M=7。该流程图中考察了 A[1：N]中所有从下标 i 到下标 j（j≥i）的各元素之和 S，并动态地记录其最大值 M。

【流程图】

希赛网在线题库



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

● 阅读以下代码，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】

```
#include<stdio.h >
void swap(int x, int y)
{
  int tmp =x;   x= y;  y= tmp;
}
int maim__(2)__
{
```

```
    int a= 3,  b= 7;
    printf("a1= %d  b1=%d\n",a,b);
    Swap( a, b);
    Printf("a2 = %d  b2=%d\n”,a,b)；
    return 0;
}
```

【代码 2】

```
#include<stdio.h>
#define  SPACE  "  //空格字符
Int main__(3)__
{
   char str[128] =" Nothing is impossible! ";
   int i,num =0,wordMark=0;

   for(i=0;str[i];i++)
      If(str[i]==SPACE)
           WordMark=0;
     else
           If(wordMark=0){
              wordMark=1;
             num++;
           }

    Printf(“%d/n”,num)
    return 0;
 }
```

【代码 3】

```
#include<stdio.h>
#define   SPACE "  //空格字符

int countStrs(char *);

int main__(4)__
{
    char str[128] = " Nothing  is  impossible! ";
    Printf("%d/n",（1）(str))
    return 0;
}

int countStrs(char *p)
{
   int num=0, wordMark= 0;
   for(;（2）; p++)   {
    If(（3）==SPACE)
        wordMark= 0;
    else
      if( !wordMark ) {
         wordMark = 1;
          ++num
}
```

希赛网在线题库

```
  }
  return （4） ;
}
```

【问题 1】（4 分）

写出代码 1 运行后的输出结果。

【问题 2】（3 分）

写出代码 2 运行后的输出结果。

【问题 3】（8 分）

代码 3 的功能与代码 2 完全相同，请补充 3 中的空缺，将解答写入答题纸的对应栏内。

● 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】

下面的程序利用快速排序中划分的思想在整数序列中找出第 k 小的元素（即将元素从小到大排序后，取第 k 个元素）。

对一个整数序列进行快速排序的方法是：在待排序的整数序列中取第一个数作为基准值，然后根据基准值进行划分，从而将待排序的序列划分为不大于基准值者（称为左子序列）和大于基准值者（称为右子序列），然后再对左子序列和右子序列分别进行快速排序，最终得到非递减的有序序列。

例如，整数序列“19, 12, 30, 11,7,53, 78, 25"的第 3 小元素为 12。整数序列“19,12,7,30,11,11,7,53,78,25,7"的第 3 小元素为 7。

函数 partition（int a[ ], int low,int high）以 a[low]的值为基准，对 a[low]、a[low+1]、…、a[high]进行划分，最后将该基准值放入 a[i] (low≤i≤high)，并使得 a[low]、a[low+1]、，．．、A[i-1]都小于或等于 a[i]，而 a[i+1]、a[i+2]、．．、a[high]都大于 a[i]。

函教 findkthElem(int a[],int startIdx,int endIdx,inr k)在 a[startIdx]、a[startIdx+1]、...、a[endIdx]中找出第 k 小的元素。

【代码】

```
#include <stdio.h>
#include <stdlib.h>

Int partition（int a [ ]，int low, int high）
{//对 a[low..high]进行划分，使得 a[low..i]中的元素都不大于 a[i+1..high]中的元素。
     int pivot=a[low];       //pivot 表示基准元素
     Int i=low,j=high;
    while(（ 1） ){
        While(i<j&&a[j]>pivot)--j;
        a[i]=a[j]
        While(i<j&&a[i]<=pivot)++i;
        a[j]=a[i]
    }
    （2）;            //基准元素定位
    return i;
}

Int findkthElem（int a[ ]，int startIdx，int endIdx, int k）
```

```
{//整数序列存储在 a[startldx..endldx]中，查找并返回第 k 小的元素。
    if (startldx<0 ||endIdx<0 || startIdx>endIdx || k<1 ||k-1>endIdx ||k-1<startIdx)
        Return-1;   //参数错误
   if(startIdx<endldx){
      int loc=partition(a, startIdx, endldx);  //进行划分，确定基准元素的位置
      if (loc==k-1)   //找到第 k 小的元素
          return （3）;
          if(k-1 <loc)     //继续在基准元素之前查找
             return  findkthElem(a, （4）,k）;
          else              / / 继续在基准元素之后查找
             return findkthElem(a，（5），k);
  }
  return a[startIdx]；
}

int main__(3)__
{
  int i, k;
  int n;
  int a[] = {19, 12, 7, 30, 11, 11, 7, 53, 78, 25, 7};

  n= sizeof(a) / sizeof(int)       //计算序列中的元素个数

  for (k=1;k＜n+1；k++){
       for(i=0;i＜n;i++){
            printf(“%d/t”,a[i]);
       }
       printf(“\n”);
       printf(“elem %d=%d\n,k,findkthElem(a,0,n-1,k));//输出序列中第 k 小的元素
}
return 0；
}
```

● 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】

图是很多领域中的数据模型，遍历是图的一种基本运算。从图中某顶点 v 出发进行广度优先遍历的过程是：

①访问顶点 v；

②访问 V 的所有未被访问的邻接顶点 W_1 ,W_2 ,..,W_k；

③依次从这些邻接顶点 W_1 ,W_2 ,..,W_k 出发，访问其所有未被访问的邻接顶点；依此类推，直到图中所有访问过的顶点的邻接顶点都得到访问。

显然，上述过程可以访问到从顶点 V 出发且有路径可达的所有顶点。对于从 v 出发不可达的顶点 u，可从顶点 u 出发再次重复以上过程,直到图中所有顶点都被访问到。

例如，对于图 4-1 所示的有向图 G，从 a 出发进行广度优先遍历，访问顶点的一种顺序为 a、b、c、e、f、d。

希赛网在线题库

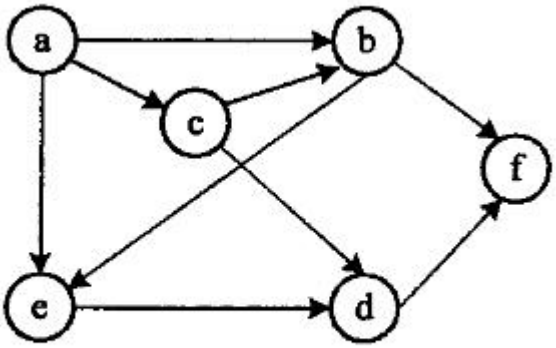


图 4-1

	a	b	c	d	e	f
a	0	1	1	0	1	0
b	0	0	0	0	1	1
c	0	1	0	1	0	0
d	0	0	0	0	0	1
e	0	0	0	1	0	0
f	0	0	0	0	0	0

图 4-2

设图 G 采用数组表示法（即用邻接矩阵 arcs 存储），元素 arcs[i][j]定义如下：

$$arcs[i][j]=\begin{cases}1 & 若G中存在边(v_i,v_j)或弧<v_i,v_j> \\ 0 & 若G中不存在边(v_i,v_j)或弧<v_i,v_j>\end{cases}$$

图 4-1 的邻接矩阵如图 4-2 所示，顶点 a~f 对应的编号依次为 0~5.因此，访问顶点 a 的邻接顶点的顺序为 b，c，e。

函数 BFSTraverse(Graph G)利用队列实现图 G 的广度优先遍历。
相关的符号和类型定义如下：

```
#define MaxN  50                    / *图中最多顶点数* /
typedef int AdjMatrix[MaxN][MaxN];

typedef struct{
   int vexnum, edgenum;          / *图中实际顶点数和边（弧）数* /
```

```
  AdjMatrix arcs;                    / *邻接矩阵* /
)Graph;
typedef int QElemType；
enum {ERROR=0;OK=1};
```

代码中用到的队列运算的函数原型如表 4-1 所述，队列类型名为 QUEUE。

表 4-1 实现队列运算的函数原型及说明

表 4-1 实现队列运算的函数原型及说明

函数原型	说明
InitQueue(QUEUE *Q)	初始化一个空队列
isEmpty(QUEUE Q)	判断队列是否为空，是则为 1，否则为 0
EnQueue(QUEUE *Q, QElemType qe)	将元素 qe 加入队列
DeQueue(QUEUE *Q, QElemType *te)	从队列头部删除元素，并通过参数 te 带回其值

【代码】

```
int BFSTraverse(Graph G)
{//对图 G 进行广度优先遍历，图采用邻接矩阵存储
  unsigned char*visited；       //visited[]用于存储图 G 中各顶点的访问标志，0 表示未访问
  int v, w, u;
  QUEUEQ  Q;

 // 申请存储顶点访问标志的空间，成功时将所申请空间初始化为 0
 visited=(char*)calloc(G.vexnum, sizeof(char));
 If( （1） )
   retum ERROR;

 （2）；          //初始化 Q 为空队列
 for( v=0; v<G.vexnum; v++){
 if(!visited[v]){        //从顶点 v 出发进行广度优先遍历
  printf("%d”，v)；       //访问顶点 v 并将其加入队列
  visited[v]=1；
  （3）；
  while(！isEmpty(Q)){
   （4）；             //出队列并用 u 表示出队的元素
  for(w=0;v<G.vexnum; w++){
    if（G.arcs[u][w]!=0&& （5）){   //w 是 u 的邻接顶点且未访问过
            printf("%d”, w)；            //访问顶点 w
            visited[w]=1；
            EnQueue(&Q, w);
     }
   }
  }
}
  free(visited);
  return OK;
)//BFSTraverse
```

● 阅读以下说明和 Java 程序，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 代码实现一个简单的聊天室系统（ChatRoomSystem），多个用户(User)可以向聊天室(ChatRoom)发送消息，聊天室将消息展示给所有用户。类图如图 5-1 所示。

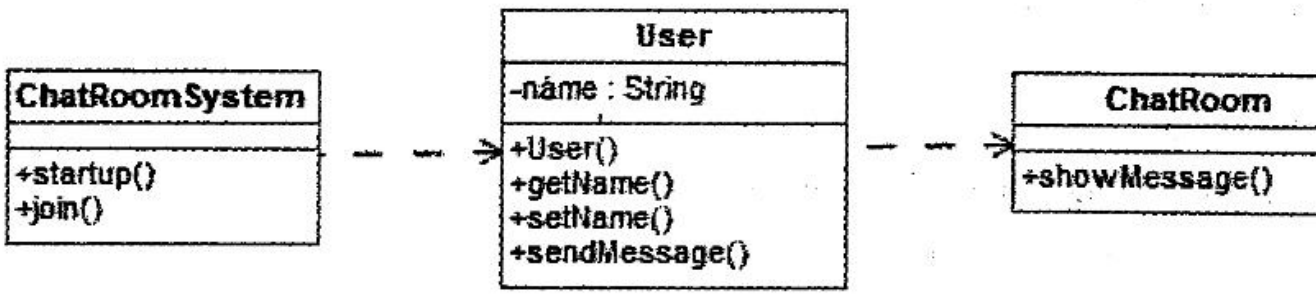


图 5-1 类图

【Java 代码】

```
class ChatRoom {
     public static void showMessage(User user, String message) {
          System.out.println("[" + user.getName__(5)__ + "] : " + message);
     }
}

class User{
   private String name;

   public String getName__(6)__ {
         return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public User(String name) {
         __（1）__=name；
}

   public void sendMessage(String message) {
         __（2）__(this, message);
  }
}

    public class ChatRoomSystem {
        public void startup__(7)__ {
           User zhang= new User("John");
           User li =new User("Leo");
```

```
            zhang.sendMessage("Hi! Leo! ");
            Li.sendMessage("Hi! John!");
    }
    public void join(User user) {
        （3）("Hello Everyone!  I am" + user.getName__(8)__);
    }
    public static void main(String[] args) {
        ChatRoomSystem crs=  （4）  ;
        Crs.startup__(9)__;
        Crs.join(  （5）  )(“Wayne”));
     }
}

/*
程序运行结果：
[John]：Hi! Leo!
[Leo]：Hi! John!
[Wayne]：Hello Everyone! I am Wayne
*/
```

● 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++代码实现一个简单的聊天室系统（ChatRoomSystem），多个用户(User)可以向聊天室（ChatRoom）发送消息，聊天室将消息展示给所有用户。类图如图 6-1 所表示。

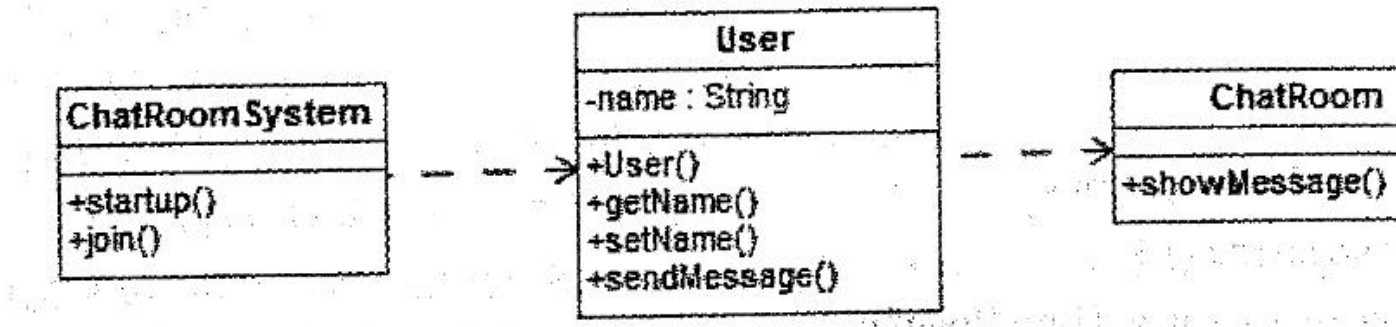


图 6-1 类图

【C++代码】

```
#include<iostream>
#include <string>

using namespace std;


class User {
private:
    string name;
```

```
public:
      User(string name){
          （1）=name；
    }
    ~User__(6)__{}
    void setName(string name) {
          this->name=name;
   }
    string getName__(7)__{
         return name;
   }

     void sendMessage(string message);
};

     class ChatRoom {      .
     public:
     static void showMessage(User* user, string message) {
             cout<<"["<<user->getName__(8)__<<"] : "<<message<<endl;
     }
};

void User::sendMessage(string message) {
           （2）(this,message);
}

class ChatRoomSystem{
public: .                              .
   void startup__(9)__ {
         User* zhang = new User(“John");
         User* li = new User("Leo");

        zhang->sendMessage("Hi! Leo!");
        li_>sendMessage("Hi! John!");
   }

   void join(User* user)  {
         （3） ("HeIIo Everyone! l am"+user->getName__(10)__);
    } .
 };

 int main__(11)__{
     ChatRoomSystem*crs=  （4） ;
     crs->startup__(12)__;
     crs->join(  （5） ("Wayne"));
     delete crs;
}
/*
程序运行结果：
[John]：Hi! Leo!
[Leo]：Hi! John!
```

[Wayne]：Hello Everyone! I am Wayne
/*

希赛网在线题库