# 华师839真题部分程序题答案

首先，华师的839试卷并没有官方的标准答案。其次，我只保证我写的答案能够顺利运行，不保证这个答案在批卷老师手里不会扣分。最后，从16年之后华师不在公布真题，所以16年之后的真题均由考生回忆所得（没有16年的真题回忆，因为谁都没有想到华师会不公布真题），向这些考生致谢。

本答案只包含部分程序题目，并非是全部内容的答案。

如果您在本答案中发现问题或有更好的建议，欢迎与我联系。

联系方式如下：

QQ 972033294

微信 dream-continue

制作者 Jerika

## 02年真题答案

### 二 L分为L1和L2该代码可在示例程序中运行

typedef struct LNode{

int data;

LNode\* next;

}LNode;

//不妨认为该线性表带头结点

//L、L1、L2分别是指向线性表头结点的指针，L1和L2初始时为NULL

void split(LNode\* L,LNode\*\* L1,LNode\*\* L2){

\*L1=(LNode\*)malloc(sizeof(LNode));

(\*L1)->next=NULL;

\*L2=(LNode\*)malloc(sizeof(LNode));

(\*L2)->next=NULL;

LNode\* LN1=\*L1,\*LN2=\*L2;

int index=0;

while(L){

if(L->data<index++){

LN1->next=(LNode\*)malloc(sizeof(LNode));

LN1=LN1->next;

LN1->data=L->data;

LN1->next=NULL;

}else{

LN2->next=(LNode\*)malloc(sizeof(LNode));

LN2=LN2->next;

LN2->data=L->data;

LN2->next=NULL;

}

L=L->next;

}

}

### 三 中后序恢复T该代码可在示例程序中运行

typedef struct TNode{

int data;

struct TNode\* lchild,\*rchild;

}TNode;

//find 从mid中找出data所在的下标

int find(int mid[],int data,int n){

for(int i=0;i<n;i++)

if(mid[i]==data)

return i;

return -1;

}

//不妨认为数组中下标从0开始存储，根节点处于post[n-1]，

//树在数字中的下标范围[0…n-1](即0<=i<n)

TNode\* createTree(int\* mid,int\* post,int n){

if(n<=0)return NULL;

TNode \*T=(TNode\*)malloc(sizeof(TNode));//新建根节点

T->data=post[n-1];

int fd=find(mid,post[n-1],n);//在mid数组中找到根节点的下标

if(fd==-1)exit(0); //数组不正确

//左子树范围mid[0…fd],post[0…fd]

T->lchild=createTree(mid,post,fd);

//右子树范围mid[fd+1…n],post[fd…n-1]

T->rchild=createTree(mid+fd+1,post+fd,n-fd-1);

return T;

}

## 03年真题答案

### 第一部分三（1） replace该代码可在示例程序中运行

int replace(char s[],char ch,char str[],char t[]){

//i:遍历s[]的下标 j:当前t[]下标

int i=0,j=0,k,sum=0;

while(s[i]){

if(s[i]-ch){

t[j++]=s[i++];

continue;

}

k=0;

while(str[k])

t[j++]=str[k++];

sum++;

i++;

}

return sum;

}

### 第一部分三（2） value9该代码无示例程序s

int value9(int d[][8],int n,int i,int j,int\* v){

if(i>0&&i<7&&j>0&&j<n-1){

\*v=0;

for(int ii=-1;ii<=1;ii++)

for(int jj=-1;jj<=1;jj++)

\*v+=d[i+ii][j+jj];

return 1;

}

return 0;

}

### 第二部分三（2） 二叉树高度和宽度该代码无示例程序s

//高度

int getHeight(TNode\* T){

return T?max(height(T->lchild),height(T->rchild))+1:-1;

}

//宽度

//T在depth层，故w[depth]++

void f(TNode\* T,int w[],int depth){

if(!T)return;

w[depth]++;

f(T->lchild,w,depth+1);

f(T->rchild,w,depth+1);

}

//宽度

int getWidth(TNode\* T){

int w[MAXN]={0};

f(T,w,0);

int maxn=0;

for(int i=0;i<MAXN;i++)

maxn=max(maxn,w[i]);

return maxn;

}

## 04年真题答案

### 第一部分三（1） data\_convert该代码无示例程序中运行

long data\_convert(long ymd){

int y=ymd/10000,m=(ymd%10000)/100,d=ymd%100,yk;

int month[]={0,31,28,31,30,31,30,31,31,30,31,30,31};

if(leap(y))month[2]=29;//如果闰年二月份加一天

yk=y\*1000;

for(int i=1;i<m;i++)

yk+=month[i];

return yk+d;

}

### 第一部分三（2） add该代码可在示例程序中运行

int add(LNUM a,LNUM b,LNUM \*c){

int i;

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

c->value[i]=a.value[i]+b.value[i];

for(i=L-1;i>0;i--)

if(c->value[i]>9){

c->value[i-1]++;//进位

c->value[i]%=10;

}

if(c->value[0]>9)return 1;//溢出

i=0;

while(!c->value[i])i++;//找到第一个不为0的数字的下标

c->length=L-i>0?L-i:1;//考虑c=0时的特殊情况

return 0;

}

## 05年真题答案

### 第一部分一 代码填空

(1).FILE\*

(2).cfp=fopen("file.dat","w")

(3).fprint(cfp,"%d %s %f",record.acctNum,record.name,record.balance)

(4).fclose(cfp)

(5).strcmp(str,password)&&++a<10

### 第一部分三 功能叙述

判断一个二进制无符号整数bits的各位相加所得的和是偶数还是奇数

### 第一部分二(3) case的奇特用法

//具体原理搜索达夫设备(Duff's Device)

\*\*\*\*\*

i=0,j=3,i=0,j=2,i=0,j=1---

i=1,j=3,i=1,j=2,i=1,j=1---

j=2,i=2,j=2,i=2,j=1---

### 第一部分四(1) “选择法” 该代码可在示例程序中运行

void sort(int\* p,int n){

if(n<=1)return;

int mini=0,tmp;

for(int i=0;i<n;i++)//找到n个中最小数的下标

if(p[mini]>p[i])

mini=i;

tmp=p[mini];

p[mini]=p[0];

p[0]=tmp;

sort(p+1,n-1);

}

### 第一部分四(2) “倒推法” 该代码无示例程序

//f(x(n))=f(x(n-1))/2-1 即f(x(n-1))=(f(x(n))+1)\*2

long f(int n,int a){

if(n<=1)return a;

return f(n-1,(a+1)\*2);

}

### 第一部分四(3) 积木分色该代码可在示例程序中运行

void swap(color\*\* a,color\*\* b){//元素为指向color的指针的指针

color \*c=\*a;

\*a=\*b;

\*b=c;

}

color inspect(color\* elem){

return \*elem;

}

void holland(color\* pc[],int n){

//rr:已分为红色块的最后一块下标 bh:已分为蓝色块的第一块下标

int rr=-1,bh=n,i=0;

while (i<bh) {

switch (inspect(pc[i])) {

case red: swap(pc+i++, pc+(++rr)); break;

case blue: swap(pc+i, pc+--bh); break;

case yellow: i++; break;

}

}

}

## 06年真题答案

### 第一部分二（1） Check该代码无示例程序

int Check(int d[],int n){

for(int i=1;i<n;i++){

if(abs(d[i])>abs(d[i-1])&&d[i]\*d[i-1]<0)continue;

return 0;

}

return 1;

}

### 第一部分二(2) Reduce该代码可在示例程序中运行

int Reduce(char s[]){

int i=0,offset=0;

while (s[i+offset]) {

if(s[i]!=s[i+1+offset]){//当前值不等于下一个，那么复制过来

s[i+1]=s[i+1+offset];

++i;

}else//偏移量+1

offset++;

}

return offset;

}

### 第二部分五 代码填空

//见蔡子经数据结构

1. <
2. ++
3. a[j]
4. j
5. t

### 第二部分七 树转邻接矩阵该代码无示例程序

typedef struct T{

int data;

struct T\* child[4];

}T;

//不妨认为初始状态下的r内元素全为0

void G2r(T\* t,int r[][12]){

if(!t)return;

for(int i=0;i<4;i++)

if(t->child[i]){

r[t->data][t->child[i]->data]=

r[t->child[i]->data][t->data]=1;

G2r(t->child[i], r);

}

}

## 07年真题答案

### 第一部分一(1) 绝对值宏该代码无示例程序

#define ABS(a,b) ((((a)-(b))>0)?((a)-(b)):((b)-(a)))

### 第一部分三(1) F10to16该代码可在示例程序中运行

void F10to16(int n){

if(n){

F10to16(n/16);

putchar(n%16>9?n%16+'A'-10:n%16+'0');

}

}

### 第一部分三(2) Goldbach该代码无示例程序

int isprime(int m){

for(int i=2;i<=(int)sqrt(m);i++)

if(!(m%i))

return 0;

return 1;

}

int Goldbach(int n){

if(n&1){

printf("%d不是偶数\n",n);

return -1;

}

int i=1,j=n-1;

while (i<=j){

if(isprime(i)&&isprime(j)){

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

return 1;

}

++i;--j;

}

printf("%d不满足Goldbach猜想\n",n);

return 0;

}

### 第一部分三(3) 数组频率最高的数及次数该代码可在示例程序中运行

int f(int a[],int n,int\* count){

\*count=0;int ans,frqNum;

for(int i=0;i<n;i++){

ans=0;

for(int j=i;j<n;j++)

if(a[i]==a[j])ans++;

if(ans>\*count){

\*count=ans;

frqNum=a[i];

}

}

return frqNum;

}

### 第二部分五 查找树层次该代码无示例程序

int getLayer(TNode\* root,int k){

if(!root)return -1;

if(root->data==k) return 0;

else if(k<root->data) return getLayer(root->lchild,k)+1;

else return getLayer(root->rchild,k)+1;

}

## 08年真题答案

### 第一部分二(1) Check该代码无示例程序

int Check(int d[],int e[],int n){

for(int i=1;i<n;i++)

if(d[i]>d[i-1]!=e[i]>e[i-1]) //形状相同则两个表达式相等

return 0;

return 1;

}

### 第一部分二(2) Go该代码无示例程序

POINT Go(POINT p0,char steps[]){

int i=0;

while (steps[i]) {

switch (steps[i++]) {

case 'E':p0.x+=steps[i++]-'0';break;

case 'W':p0.x-=steps[i++]-'0';break;

case 'N':p0.y+=steps[i++]-'0';break;

case 'S':p0.y-=steps[i++]-'0';break;

}

}

return p0;

}

### 第二部分三(1) 有序链表去重该代码可在示例程序中运行

//不妨认为该线性不带头结点

void RemoveDuplicates(LINK\* head){

if(!head)return;

LINK\* tmp;

while (head->next) {

if(head->next->data==head->data){

tmp=head->next;

head->next=tmp->next;

free(tmp);

}else head=head->next;

}

}

### 第二部分三(2) 顺序存贮树遍历该代码可在示例程序中运行

//不妨认为T的数组从下标1开始存储,即T[0]内不存放数据,范围T[1...n]

void midTraverse(int T[],int cur,int n){

if(cur>n)return;

midTraverse(T, cur\*2, n);

printf("%d ",T[cur]);

midTraverse(T, cur\*2+1, n);

}

## 09年真题答案

### 第一部分三(1) Check该代码无示例程序

类似07第一部分三(1)

### 第一部分四(1) 文件输出该代码可在示例程序中运行

int main(int argc,char\* argv[]){

if(argc<2)exit(0);

FILE\* fp=fopen(argv[1],"r");

if(fp==NULL)exit(0);

char buf[17];

long offset=ftell(fp);

while (fgets(buf, 17, fp)!=NULL) {

printf("%08x ",offset);

offset=ftell(fp);

for(int i=0;i<strlen(buf);i++)

printf("%02x ",buf[i]);

for(int i=strlen(buf);i<17;i++)

printf(" ");

for(int i=0;i<strlen(buf);i++)

putchar(isprint(buf[i])?buf[i]:'.');

putchar('\n');

}

}

### 第一部分四(2) 是非题该代码可在示例程序中运行

int p(int\* a,int\* b,int n){

int sum=0;

for(int i=0;i<n;i++)

if(a[i]==b[i])

sum++;

return sum;

}

int main(){

int a[10]={0,1,1,1,0,1,0,0,1,0},

b[10]={0,0,1,1,1,0,1,1,0,0},

c[10]={1,0,1,0,1,1,1,0,1,1};

for(int i=0;i<10;i++){

a[i]=!a[i];

for(int j=i+1;j<10;j++){

a[j]=!a[j];

for(int k=j+1;k<10;k++){

a[k]=!a[k];

if(p(a,b,10)==7&&p(a,c,10)==7){

printf("Answer:\n");

for(int m=0;m<10;m++){

printf("%d ",a[m]);

}

return 0;

}

a[k]=!a[k];

}

a[j]=!a[j];

}

a[i]=!a[i];

}

}

### 第二部分三 代码填空

//见蔡子经数据结构

1. –-
2. >
3. 0
4. >
5. i
6. n

### 第二部分四 代码填空

//见蔡子经数据结构

1. 1
2. rchild
3. ==0
4. lchild

### 第二部分七 x为y、z共同祖先该代码可在示例程序中运行

首先找到x的结点，之后从以x结点为根节点找y和z

T\* f(T\* t,int x){

if(!t)return NULL;

if(t->data==x)return t;

T\* ft=f(t->lchild,x);

return ft?ft:f(t->rchild,x);

}

int isCommonAncestor(T\* t,int x,int y,int z){

T\* fx=f(t,x);

if(fx)

return f(fx,y)&&f(fx,z);

}

## 10年真题答案

### 第一部分三(1) ui2hex该代码可在示例程序中运行

char\* ui2hex(unsigned n,char \*s){

if(n==0){s="0";return s;}

int i=0;

while (n) {//逆序的hex

s[i++]=n%16>9?n%16-10+'a':n%16+'0';

n/=16;

}

for (int j=0;j<i/2;j++) {//逆置数组

s[i]=s[j];

s[j]=s[i-j-1];

s[i-j-1]=s[i];

}

s[i]='\0';

return s;

}

### 第一部分四(1) 组合排列该代码无示例程序但可直接运行,注意指定输入输出文件

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAXN 10005

//不妨认为文件格式为id name score\n

typedef struct student{

char id[15];

char name[15];

float score;

}student;

student stus[MAXN];

int cmp(const void\* stua,const void\* stub){

return strcmp(((student\*)stua)->id,((student\*)stub)->id);

}

int main(int argc,char\* argv[]){

if(argc<3)exit(0);

FILE\* fin=fopen(argv[1],"r"),

\*fout=fopen(argv[2],"w");

if(!fin||!fout)exit(0);

int n=0;

while (fscanf(fin,"%s%s%f",stus[n].id,

stus[n].name,&stus[n].score)!=EOF)

++n;

fclose(fin);

int sapSize=n\*0.01;

for(int i=0;i<sapSize;i++){

//从i...n-1随机一个下标,交换该信息与第i个信息

//避免随机数出现相同的情况

int sapIdx=(rand()%(n-i)+i);

stus[n]=stus[sapIdx];//交换

stus[sapIdx]=stus[i];

stus[i]=stus[n];

}

qsort(stus, sapSize, sizeof(student), cmp);

for(int i=0;i<sapSize;i++)

fprintf(fout, "%s %s %f\n",

stus[i].id,stus[i].name,stus[i].score);

fclose(fout);

}

### 第一部分四(2) 组合排列该代码可在示例程序中运行

//a[0...n-1]的m个组合排列,当前处于第i个下标,组合排列中已有k个,b储存组合排列,b[0...k-1],初始时i=k=0

void comb(char a[],int n,int m,int i,int k,char b[]){

if(k==m){//完成组合排列

b[k]='\0';

puts(b);

return;

}

if(n-i+1<=m-k)return;//剩余元素过少，不能完成排列

b[k]=a[i];

comb(a,n,m,i+1,k+1,b); //a[i]加入组合后递归

comb(a,n,m,i+1,k,b); //a[i]不加入组合后递归

}

### 第二部分二 代码填空

//见蔡子经数据结构

1. >
2. >=
3. >=
4. n+k
5. i+k

### 第二部分八 代码填空

//见蔡子经数据结构

1. 1
2. head[u]
3. 0
4. t->ver
5. t->link

## 11年真题答案

### 六 floyd代码填空

//见蔡子经数据结构

1. <a[i][j]
2. a[i][k]+a[k][j]
3. k
4. print\_path(path[i][k],k)
5. print\_path(k,path[k][j])

### 七 打印非叶结点该代码无示例程序

void print(T\* t){

if(!t)return;

if(!t->lchild&&!t->rchild) printf("%d ",t->data);

print(t->lchild);

print(t->rchild);

}

### 十(1) i2ba该代码可在示例程序中运行

unsigned char\* i2ba(int n,unsigned char\* s){

for(int i=0;i<sizeof(n)/sizeof(s[0]);i++,n>>=8)

s[i]=n&(unsigned char)-1;//(unsigned char)-1-->(0xff)

return s;

}

### 十(2) maxBlock该代码可在示例程序中运行

const int steps[4][2]={0,1,0,-1,1,0,-1,0};

int dfs(int \*\*a,int i,int j,int h,int w){

a[i][j]=0;

int sum=0;

for(int k=0;k<4;k++){

int ii=i+steps[k][0],jj=j+steps[k][1];

if(ii>=0&&ii<h&&jj>=0&&jj<w&&a[ii][jj])

sum+=dfs(a,ii,jj,h,w); //dfs

}

return sum+1;

}

int maxBlock(int \*\*a,int h,int w){

int maxn=0;

for(int i=0;i<h;i++)

for(int j=0;j<w;j++)

if(a[i][j]) maxn=max(maxn,dfs(a,i,j,h,w));

return maxn;

}

### 十一(1) 不含101子串个数该代码无示例程序但可直接运行

#include <stdio.h>

int f(int n){

int dp[22];

dp[1]=2;dp[2]=4;dp[3]=7;

for(int i=4;i<=n;i++)

dp[i]=dp[i-1]\*2-dp[i-2]+dp[i-3]; //递推公式

return dp[n];

}

int main(){

int n;scanf("%d",&n);printf("%d:%d\n",n,f(n));

}

### 十一(2) 解一元一次方程

//变态题立即推—->放弃考研

## 12年真题答案

### 第一部分三(1) Insert该代码可在示例程序中运行

NODE\* Insert(NODE \*head,NODE \*p){

if(!head||head->data>p->data){

p->next=head;

return p;

}

NODE\* node=head;

while (node){

if(!node->next||node->next->data>p->data){

p->next=node->next;

node->next=p;

break;

}

node=node->next;

}

return head;

}

### 第一部分三(2) PRINT该代码可在示例程序中运行

void ui2n(unsigned value,int n){ //十进制转换为n进制(n=2,8,10,16)

if (value) {

ui2n(value/n, n);

putchar(value%n>9?value%n-10+'a':value%n+'0');

}

}

void PRINT(char fmt[],unsigned value){

int i=0;

while (fmt[i]) {

if(fmt[i]!='%'){

putchar(fmt[i++]);

continue;

}

switch (fmt[++i]) {

case 'b': ui2n(value, 2); break;

case 'u': ui2n(value, 10); break;

case 'x': ui2n(value, 16); break;

case 'o': ui2n(value, 8); break;

}

++i;

}

}

### 第一部分四 杨辉三角该代码无示例程序但可直接运行

#include <stdio.h>

#define MAXN 52

long long c[MAXN][MAXN];

void init(){

for(int i=0;i<MAXN;i++)

c[i][0]=c[i][i]=1;

for(int i=2;i<MAXN;i++)

for(int j=1;j<i;j++)

c[i][j]=c[i-1][j-1]+c[i-1][j];

}

int main(){

init();

int n,k;

label: scanf("%d%d",&n,&k);

if(n>0&&n<=50&&k>0&&k<=n)printf("C(%d,%d)=%lld",n,k,c[n][k]);

else {

puts("Enter n(1<=n<=50),k(1<=k<=n)");

goto label;

}

}

### 第二部分七(1) 层次遍历该代码无示例程序

//(1) 根节点入队

//(2) 队不空则出队t

//(3) 将t的子节点入队

//(4) 重复步骤(2),(3)直到队空

void layerTraverse(T\* t){

if(!t) return;

Queue q[MAXN];

q.enqueue(t);

while (!q.empty()) {

t=q.dequeue();

visit(t);

if(t->lchild)q.enqueue(t->lchild);

if(t->rchild)q.enqueue(t->rchild);

}

}

### 第二部分七(2) 判断联通该代码无示例程序

typedef struct L\_NODE{//邻接表结点

int ver;

struct L\_NODE\* link;

}L\_NODE;

L\_NODE\* head[5];

int vis[5]={0};

//初始v=0,若连通返回5,联不通返回其他数值

int dfs(int v){

int sum=0;

L\_NODE \*t=head[v];

while (t) {

if(!vis[t->ver]) sum+=dfs(t->ver);

t=t->link;

}

return sum+1;

}

## 13年真题答案

### 第一部分三(3) bsearch该代码可在示例程序中运行

void\* bsearch(void\* key,void\* base,size\_t num,size\_t width,

int (\*compare)(const void\* elem1,const void\* elem2)){

size\_t low=0,high=num-1,mid;

while(low<=high) {

mid=(low+high)>>1;

if(compare(key,(char\*)base+mid\*width)==0)

return (char\*)base+mid\*width;

else if(compare(key,(char\*)base+mid\*width)<0)

high = mid-1;

else

low = mid+1;

}

return NULL;

}

### 第一部分四 MCNuggests该代码无示例程序

//考虑6,9的组合，对于所有3m(m>1)的n,都可以被6和9表出

//那么,当加入20后,对所有的n>=60,都可以被6,9,20表出//(61=21+40,62=42+20,依次类推...)

//结论:n>=60的数都可以被6,9,20表出,n<6的数都不能表出,

//所以只需判断6<n<60的数,对于n-20k>=6的数若是3的倍数,则可表出

int MCNuggests(int n){

if(n<6)return 0;

if(n>=60)return 1;

if(n%3==0)return 1;

for(int k=0;n-k\*20>=6;k++)

if((n-k\*20)%3==0)

return 1;

return 0;

}

### 第一部分五 合法身份证该代码无示例程序但可直接运行

#include <string.h>

#include <stdio.h>

const int w[]={7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2};

const char t[]="10X98765432";

int islegal(char\* s){

if(strlen(s)!=18)return 0;

int sum=0;

for(int i=0;i<17;i++)

if(s[i]<'0'||s[i]>'9') return 0;

else sum+=(s[i]-'0')\*w[i];

return s[17]==t[sum%11];

}

int main() {

char s[100];

scanf("%s",s);

printf("%s is %s\n",s,islegal(s)?"legal":"illegal");

}

### 第二部分十三 后序遍历5次树该代码无示例程序

typedef struct T{

int data;

struct T\* child[5];

}T;

void postTraverse(T\* t){

if(!t)return;

for(int i=0;i<5;i++)

postTraverse(t->child[i]);

visit(t);

}

## 14年真题答案

### 第一部分四split该代码可在示例程序中运行

//s:原字符串

//c:作为分隔的字符

//result:储存分隔完成后的字符串数组 是char\*[]的地址,为了当result数组为空时不出异常

//return:分隔后的数组元素个数,result[0...n-1]

int split(char\* s,char c,char \*\*\*result){

if(!\*result)\*result=(char\*\*)malloc(sizeof(char\*)\*500);

char tmp[1001];

int i=0,j=0,n=0;

while (s[i]) {

if(s[i]!=c){

tmp[j++]=s[i++];

continue;

}

if(j){

tmp[j++]=0;

(\*result)[n]=(char\*)malloc(sizeof(char)\*j);

strcpy((\*result)[n++], tmp);

j=0;

}

++i;

}

if(j){

tmp[j++]=0;

(\*result)[n]=(char\*)malloc(sizeof(char)\*j);

strcpy((\*result)[n++], tmp);

}

return n;

}

### 第一部分五 文件操作程序该代码无示例程序但可直接运行,注意输入指定文件

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

int hash(char\* s){

int i=0,sum=0;

while(s[i])

sum+=s[i++];

return sum%10007;

}

int cmp(const void\* a,const void\* b){

char\* x=\*((char\*\*)a),\*y=\*((char\*\*)b);

if(hash(x)==hash(y))

if(strlen(x)==strlen(y))

return strcmp(y,x);

else

return strlen(y)-strlen(x);

else

return hash(x)-hash(y);

}

int main(int argc,char\*\* argv){

if(argc<1)exit(0);

int n=atoi(argv[1]);

FILE\* fin=fopen("data.in","r"),

\*fout=fopen("data.out","w");

if(!fin||!fout)exit(0);

char tmp[10001];

char\*\* strs=(char\*\*)malloc(sizeof(char\*)\*n);

for(int i=0;i<n;i++){

fgets(tmp,10001,fin);

strs[i]=(char\*)malloc(sizeof(char)\*(strlen(tmp)+1));

strcpy(strs[i], tmp);

}

fclose(fin);

qsort(strs, n, sizeof(strs[0]), cmp);

for(int i=0;i<n;i++){

fputs(strs[i], fout);

free(strs[i]);

}

fclose(fout);

free(strs);

}

### 第二部分八 代码填空

//见蔡子经数据结构

1. >=
2. <=j
3. k++
4. k
5. k

### 第二部分十二 代码填空

//见蔡子经数据结构

1. !=
2. \*p\_q
3. lchild
4. rchild

## 15年真题答案

### 第一部分三 Merge该代码无示例程序但可直接运行,注意输入指定文件

//从尾部开始的归并排序

void Merge(int\* A,int n,int\* B,int m){

int r=n+m-1;

--n;--m;

while (n>=0||m>=0)

A[r--]=A[n]>B[m]?A[n--]:B[m--];

while (m>=0)

A[r--]=B[m--];

}

### 第一部分四 单向链表去重该代码可在示例程序中运行

typedef struct LINK{

int data;

struct LINK\* next;

}LINK;

//不妨认为该线性带头结点

void RemoveDuplicates(LINK\* head){

if(!head||!(head->next))return;

head=head->next;

LINK\* work,\*tmp;

while (head) {//删除所有和head值相等后续的结点

work=head;//work为待判断结点的前驱

while (work->next)

if(work->next->data==head->data){

//work->next和head值一样，那么删除work->next

tmp=work->next;

work->next=tmp->next;

free(tmp);

}else work=work->next;

head=head->next;

}

}

### 第一部分五 数独该代码无示例程序但可直接运行,注意输入指定文件

#include <stdlib.h>

#include <string.h>

#include <stdio.h>

int a[40][40];int vis[40];

int main(){

FILE\* fp=fopen("dataIN","r");

if(!fp)exit(0);

int n=fscanf(fp,"%d",&n);

for(int i=0;i<n\*n;i++)

for(int j=0;j<n\*n;j++){

fscanf(fp,"%d",a[i]+j);

if(a[i][j]<1||a[i][j]>n\*n) goto label;

}

fclose(fp);

for(int i=0;i<n\*n;i++){

memset(vis, 0, sizeof(vis));

for(int j=0;j<n\*n;j++)//行

if(vis[a[i][j]])goto label;

else vis[a[i][j]]=1;

memset(vis, 0, sizeof(vis));

for(int j=0;j<n\*n;j++)//列

if(vis[a[j][i]])goto label;

else vis[a[j][i]]=1;

}

for(int i=0;i<n;i++)

for(int j=0;j<n;j++){//小矩阵

memset(vis, 0, sizeof(vis));

for(int k=0;k<n;k++)

for(int m=0;m<n;m++)

if(vis[a[i\*n+k][j\*n+m]])goto label;

else vis[a[i\*n+k][j\*n+m]]=1;

}

puts("Yes");

return 0;

label: puts("No");

return 0;

}

### 第二部分十 扩充标准型该代码无示例程序

typedef struct KT{

int data;

struct KT\* lchild,\*rchild,\*parent;

}KT;//扩充标准型

typedef struct T{

int data;

struct T\* lchild,\*rchild;

}T;//标准型

//t为二叉树根节点,parent为应是t的父节点的KT,初始时parent为NULL

KT\* T2KT(T\* t,KT\* parent){

if(!t)return NULL;

KT\* node=(KT\*)malloc(sizeof(KT));

node->parent=parent;

node->lchild=T2KT(t->lchild,node);

node->rchild=T2KT(t->rchild,node);

return node;

}

## 17年真题回忆版答案

### 第一部分一 fun

不报错会运行直到内存满

### 第一部分四 MYprintf该代码可在示例程序中运行

#include <stdarg.h>//可变参数头文件

void ui2n(unsigned value,int n){ //十进制转换为n进制(n=2,8,10,16)

if (value) {

ui2n(value/n, n);

putchar(value%n>9?value%n-10+'a':value%n+'0');

}

}

void MYprintf(char fmt[],...){

va\_list ap; //1🡪声明可变参数类型ap

va\_start(ap,fmt); //2🡪初始化ap

int i=0;

while (fmt[i]) {

if(fmt[i]!='%'){

putchar(fmt[i++]);

continue;

}

switch (fmt[++i]) {

//3🡪从ap中取值

case 'd': printf("%d",va\_arg(ap,int)); break;

case 'b': ui2n(va\_arg(ap, unsigned),2);break;

case 'o': ui2n(va\_arg(ap, unsigned),8);break;

case 'x': ui2n(va\_arg(ap, unsigned),16);break;

case 'c': putchar(va\_arg(ap, char)); break;

case 's':{

char\* s=va\_arg(ap, char\*);

int j=0;

while (s[j])

putchar(s[j++]);

break;

}

default: putchar(fmt[i]);

break;

}

++i;

}

va\_end(ap); //4🡪销毁ap

}

### 第一部分五 文件操作矩形该代码无示例程序但可直接运行,注意输入指定文件

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

#define N 10005

typedef struct point{

float x1,y1,x2,y2;

}point;

float area(point a){

return abs((a.x2-a.x1)\*(a.y2-a.y1));

}

float dis2x(point a){

return a.x1\*a.x1+a.y1\*a.y1;

}

float dis2y(point a){

return a.x2\*a.x2+a.y2\*a.y2;

}

int cmp(const void\* a,const void\*b){

point x=\*((point\*)a),y=\*((point\*)b);

if(area(x)==area(y))

if(dis2x(x)==dis2x(y))

return dis2y(x)-dis2y(y);

else

return dis2x(x)-dis2x(y);

else

return area(x)-area(y);

}

int isexist(point ps[],int n,point p){

for(int i=0;i<n;i++)

if(ps[i].x1==p.x1&&ps[i].x2==p.x2&&

ps[i].y1==p.y1&&ps[i].y2==p.y2)

return 1;

return 0;

}

int main(){

FILE\* fin=fopen("data.in", "r"),

\*fout=fopen("data.out", "w");

if(!fin||!fout)exit(0);

int n,x1,y1,x2,y2;

point ps[N];

while (fscanf(fin, "%f%f%f%f",&ps[n].x1,&ps[n].y1,

&ps[n].x2,&ps[n].y2)!=EOF)

if(!isexist(ps, n, ps[n]))n++;

fclose(fin);

qsort(ps, n, sizeof(ps[0]), cmp);

for(int i=0;i<n;i++)

fprintf(fout, "%f %f %f %f\n",ps[i].x1,ps[i].y1,

ps[i].x2,ps[i].y2);

fclose(fout);

}

### 第二部分九 堆判断该代码无示例程序

//不妨认为是大顶堆,那么只需判断每个根节点是否比其叶节点大

//注意一般认为的堆都是完全二叉树,因为是回忆版所以不知道有没有这个条件

int isheap(T\* t){

if(!t)return 1;

if((!t->lchild||t->data>t->lchild->data)&&

(!t->rchild||t->data>t->rchild->data))

return isheap(t->lchild)&&isheap(t->rchild);

return 0;

}

### 第二部分十 长为k的路径该代码无示例程序

typedef struct L\_NODE{

int vec;

struct L\_NODE\* link;

}L\_NODE;

L\_NODE\* head[N];

int vis[N]={0};

//a->b是否有长为k的路径

int judge(int a,int b,int k){

if(a==b&&k==0)return 1;

if(k<0)return 0;

vis[a]=1;

for(L\_NODE\* v=head[a];v;v=v->link){

//v->b是否有长为k-1的路径

if(!vis[v->vec]&&judge(v->vec, b, k-1))

return 1;

}

vis[a]=0; //恢复标记

return 0;

}

## 18年真题回忆版答案

### 第一部分二 p位到q位该代码可在示例程序中运行

char\* p2q(unsigned n,int p,int q){

char\* s=(char\*)malloc(sizeof(char)\*(q-p+2));

int i=q;

while (i>=p) {

s[q-i]=((n>>i)&1)+'0';

i--;

}

s[q-i]=0;

return s;

}

### 第一部分三 实现一个排序算法该代码无示例程序

//懒得写网上抄的,void指针使用时注意转换,快排忘记了可以用其他简单的算法

//不用看个这答案

void swap(const void\* a, const void\* b, int size){

char tmp = 0;

int i = 0;

while (size > 0) {

tmp = \*((char\*)a + i);

\*((char\*)a + i) = \*((char\*)b + i);

\*((char\*)b + i) = tmp;

++i;

--size;

}

}

void Qsort(void\* base, int left, int right, int size, int (\*cmp)(const void\* a, const void\* b)){

if (left >= right) return;

char\* pleft = (char\*)base + left \* size;

char\* pkey = (char\*)base + (left + (right - left) / 2) \* size;

swap(pleft, pkey, size);

int last = left;

char\* plast = (char\*)base + last \* size;

for (int i = left + 1; i <= right; ++i) {

char\* pi = (char\*)base + i \* size;

if (cmp(pi, pleft) < 0) {

++last;

plast = (char\*)base + last \* size;

swap(pi, plast, size);

}

}

swap(pleft, plast, size);

Qsort(base, left, last - 1, size, cmp);

Qsort(base, last + 1, right, size, cmp);

}

### 第一部分四 立方根该代码可在示例程序中运行

//可以通过二分法,注意正负以及0...1和1...n的情况

double cub(double n){

double eps=1e-4,low,high,mid;

int flag=n>0?1:-1;

n=abs(n);

if(n>1)low=1,high=n;

else low=n,high=1;

mid=(low+high)/2;

while (abs(mid\*mid\*mid-n)>eps) {

mid\*mid\*mid>n?high=mid:low=mid;

mid=(low+high)/2;

}

return mid\*flag;

}

//以下通过牛顿迭代法

double cub(double n){

double eps=1e-4,r=n;

while (abs(r\*r\*r-n)>eps) {

r=2\*r/3+n/3/r/r;

}

return r;

}

### 第一部分五 文件操作该代码无示例程序

#include <stdlib.h>

#include <stdio.h>

#define MAXN 10005

typedef struct student{

long id;

char name[20];

float score;

}student;

student stus[MAXN];

int cmp(const void\* a,const void\* b){

student x=\*((student\*)a),y=\*((student\*)b);

if(x.score==y.score)return y.id-x.id;

return x.score-y.score;

}

int main(int argc,char\*\* argv){

if(argc<4)exit(0);

FILE\* fin1=fopen(argv[1], "r"),

\*fin2=fopen(argv[2], "r"),

\*fout=fopen(argv[3], "w");

if(!fin1||!fin2||!fout)exit(0);

int n=0;

while(fscanf(fin1, "%ld%s%f",&stus[n].id,

stus[n].name,&stus[n].score)!=EOF)

n++;

fclose(fin1);

while(fscanf(fin2, "%ld%s%f",&stus[n].id,

stus[n].name,&stus[n].score)!=EOF)

n++;

fclose(fin2);

qsort(stus, n, sizeof(stus[0]), cmp);

for(int i=0;i<n;i++)

fprintf(fout, "%ld %s %f\n",stus[i].id,

stus[i].name,stus[i].score);

fclose(fout);

}

### 第二部分六 奇偶划分该代码可在示例程序中运行

void split(int a[],int n){

int h=0,r=n-1,t;

while (h<=r)

if(a[h]&1)h++;

else {

t=a[h];

a[h]=a[r];

a[r]=t;

r--;

}

}

### 第二部分八 祖先打印该代码无示例程序

int printAncestors(T\* t,int x){

if(!t)return 0;

if(t->data==x)return 1;

if(printAncestors(t->lchild, x)){

printf("%d ",t->data);

return 1;

}else if(printAncestors(t->rchild, x)){

printf("%d ",t->data);

return 1;

}

return 0;

}