

**厦门大学《C语言程序设计》课程试卷**

**信息 学院＿＿系 2021 年级 计算机类 专业**

**学年学期：21221 主考教师：黄绍辉 等** **B卷（√）**

**一、写出下列程序段的运行结果（40分）**

**1.（4分）**

**int x[3][2]={0}, i;**

**for(i=0;i<3;i++)**

**scanf("%d",x[i]);**

**printf("%d,%d,%d,%d",**

**x[0][0],x[0][1],**

**x[0][2],x[1][0]);**

**若运行时输入：1 2 3<回车>**

**2.（4分）**

**int x=1,a=0,b=0;**

**switch(x) {**

**case 0: b++;**

**case 1: a++;**

**case 2: a++; b++;**

**}**

**printf("%d,%d",a,b);**

**3.（4分）**

**float x=2.0,y=4.0;**

**if (x<0.0) y=3.0;**

**else if (x<10.0) y=x/2;**

**else y=2.0;**

**printf("%f\n",y);**

**int a=-1,b=1,k;**

**if ((++a<0) && !(b--<=0))**

**printf("%d,%d",a,b);**

**else**

**printf("%d,%d",a,b);**

**4.（4分）**

**int k=5;**

**while (k=0) k=k-3;**

**printf("%d\n", k);**

**int a=1,b=10;**

**do {**

**b-=a;**

**a++;**

**} while (b-- < 0);**

**printf("%d,%d",a,b);**

**5.（4分）**

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

**printf("%d", a[1][2]);**

**char s[6]={'a','b',0,'c','d','\0'};**

**printf("%s", s);**

**6.（4分）**

**char a[]="abcd", b[]="123";**

**strcpy(a,b);**

**printf("%s,",a);**

**char s[3][10]={"abcd", "abc", "ab"};**

**if (s[0] > s[1])**

**printf("%s", s[0]);**

**else**

**printf("%s", s[1]);**

**7.（4分）**

**int f1()**

**{**

**static int k=1;**

**return ++k;**

**}**

**int g=8;**

**int f2()**

**{**

**int g=5;**

**return ++g;**

**}**

**int main()**

**{**

**int i;**

**for(i=1;i<=3;i++) f1();**

**printf("%d, %d", f1(), f2());**

**return 0;**

**}**

**8.（4分）**

**void f(char ch)**

**{**

**if (ch>'A') f(ch-1);**

**putchar(ch);**

**}**

**void g(char ch)**

**{**

**if (ch>'B') g(ch-1);**

**putchar(ch);**

**if (ch>'B') g(ch-2);**

**}**

**int main()**

**{**

**f('D');**

**putchar('\n');**

**g('D');**

**return 0;**

**}**

**9.（4分）**

**struct stu {**

**int num;**

**int age;**

**};**

**struct stu t[3] = {{101,18}, {102,21}, {103,19}};**

**struct stu \*p=t,\*q=t;**

**printf("%d,%d", (p++)->num, (\*++q).age);**

**10.（4分）**

**FILE \*fp;**

**long d=20211224;**

**char c1,c2;**

**char str[4];**

**fp=fopen("a.tmp","w");**

**fprintf(fp,"%ld", d);**

**fclose(fp);**

**fp=fopen("a.tmp","r");**

**c1=fgetc(fp);**

**fgets(str, 4, fp);**

**c2=fgetc(fp);**

**fscanf(fp, "%d", &d);**

**fclose(fp);**

**printf("%c,%s,%c,%d",**

**c1,str,c2,d);**

**二、改错题（20分）**

**以下程序实现：从键盘输入30个学生的成绩数据，将这些数据按照成绩从低到高排序后保存到磁盘文件“stu.dat”上。程序中共有11个错误，指出所有错误所在位置并改正。**

**#include <stdio.h> /\*第 1行\*/**

**#define SIZE 30 /\*第 2行\*/**

**struct STU /\*第 3行\*/**

**{ /\*第 4行\*/**

**char name[10]; /\*第 5行\*/**

**double score; /\*第 6行\*/**

**} /\*第 7行\*/**

**void save(struct STU stud[SIZE]) /\*第 8行\*/**

**{ /\*第 9行\*/**

**File \*fp; /\*第10行\*/**

**int j; /\*第11行\*/**

**if((fp=fopen("d:\\stu.dat","wb"))==NULL) /\*第12行\*/**

**{ /\*第13行\*/**

**printf("cannot open file\n"); /\*第14行\*/**

**return; /\*第15行\*/**

**} /\*第16行\*/**

**for(i=0;i<SIZE;i++) /\*第17行\*/**

**if(fread(&stud[i],sizeof(struct STU),1,fp)!=1)/\*第18行\*/**

**printf("file write error\n"); /\*第19行\*/**

**} /\*第20行\*/**

**void sort(struct STU stud[SIZE]) /\*第21行\*/**

**int main() /\*第22行\*/**

**{ /\*第23行\*/**

**struct STU stud[SIZE]; /\*第24行\*/**

**int i; /\*第25行\*/**

**for(i=0;i<SIZE;i++) /\*第26行\*/**

**{ /\*第27行\*/**

**scanf("%s%lf",stud[i].name, stud[i].score);/\*第28行\*/**

**} /\*第29行\*/**

**sort(stud); /\*第30行\*/**

**save(stud[size]); /\*第31行\*/**

**return 0; /\*第32行\*/**

**} /\*第33行\*/**

**void sort(struct STU stud[SIZE]) /\*第34行\*/**

**{ /\*第35行\*/**

**STU temp; /\*第36行\*/**

**int i,j,k; /\*第37行\*/**

**for(i=0;i<SIZE-1;i++) /\*第38行\*/**

**{ k=i; /\*第39行\*/**

**for(j=i+1;j< SIZE;j++) /\*第40行\*/**

**if(stud[j].score> stud[k].score) /\*第41行\*/**

**k=j; /\*第42行\*/**

**temp=stud[k]; /\*第43行\*/**

**stud[k]= stud[i]; /\*第44行\*/**

**stud[i] = temp; /\*第45行\*/**

**} /\*第46行\*/**

**return 0; /\*第47行\*/**

**} /\*第48行\*/**

**三、编程题（40分） 注意：程序中请添加必要的注释**

**1．（12分）输入一个已降序排列的整型数组（先输入一个整数表示数组大小，数组大小不超过1000，再依次输入数组各元素），最后输入一个整数，插入到相应位置，显示插入数据后的数组，各元素间用逗号分隔。**

**2．（14分）某图书馆的借阅记录采用数组存储，每条借阅记录包含：学工号（no整型），借阅的书名(bname,字符串类型，长度不超过32)，借阅日期 (bdate，结构体类型Date)等信息，请要求完成：**

**1）请写出BorrowedRecord的定义，用于存储借阅记录信息。假设Date类型已经定义,可以直接使用，Date结构体包含三个整型成员year, month和day。**

**2）请设计一个排序函数，对借阅记录数组按照借书日期的先后顺序排列，越早借阅的记录排在越前面，函数原型如下所示；**

**void Sort(struct BorrowedRecord brecs[ ] , int N)**

**其中，brecs为借阅信息数组，N为借阅记录数;**

**3）假设借阅时长≤60天，请设计一个查询函数，查找在指定的日期当日已经超期的所有借阅记录，并打印输出。函数的原型如下所示：**

**void QueryExpire(struct BorrowedRecord brecs[ ], int N, Date qrydate)**

**其中，参数brecs和 N的含义与（2）相同，qrydate为查询指定的日期。**

**查询结果的输出格式如下：**

[1] 1001 计算机组成原理 2021年5月7日

[2] 1003 C语言程序设计实践 2021年6月3日

**3． 用链表存储一组类型为int的**整数**。下图所示是4个整数20，21，21，20被存储在链表A的情形：**

**A**

**20**

**21**

**21**

**20**

**^**

**根据以上描述，完成以下问题：**

**(1) 规定两个成员名必须为d和next，写出链表结点struct N的定义；（2分）**

**(2) 编写函数：int sym(struct N \*head)，判断非空链表head是否为对称链表，是则返回1，否则返回0。假设链表长度不超过1000；（5分）**

**(3) 编写函数left，形参是A，功能是按照结点出现顺序复制对称链表A，但是仅仅复制左半边(含正中间的结点)，函数返回新链表B，原链表A不允许做任何修改。例如执行B=left (A)后，得到的新链表B如下：（7分）**

**B**

**20**

**21**

**^**

**参考答案**

**一、（40分）评分标准：每小题4分**

|  |  |  |
| --- | --- | --- |
| 题号 | 答案 | 备注 |
| **1** | **1,0,2,2** | **每个数字正确得1分** |
| **2** | **2,1** | **每项2分** |
| **3** | **1.000000**  **0,1** | **第1行2分**  **第2行每项1分** |
| **4** | **0**  **2,8** | **第1行2分**  **第2行每项1分** |
| **5** | **6,ab** | **每项2分** |
| **6** | **123,abc** | **每项2分** |
| **7** | **5,6** | **每项2分** |
| **8** | **ABCD**  **BCADB** | **每行2分** |
| **9** | **101,21** | **每项2分** |
| **10** | **2,021,1,224** | **每项1分** |

**二、改错题（20分）**

**共11个错，找到给1分，修改正确给1分，满分20，即允许一个错误没找到**

**1. 第7行，加；**

**2. 第10行，File改为 FILE**

**3. 第11行，j 改为 i**

**4. 第18行，fread 改为 fwrite**

**5. 第19行后增加fclose（fp）**

**6. 第21行后增加 ;**

**7. 第28行，stud[i].score前加&**

**8. 第31行，去掉[size]**

**9. 第36行，增加struct**

**10. 第41行，>改为<**

**11. 第47行，删除 0; 或 整行删除**

**三、编程题**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{ int n,k,i;**

**scanf("%d",&n);**

**int a[n+1]; //定义数组，2分**

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

**scanf("%d",&a[i]);**

**scanf("%d",&k); //输入数据，1分**

**for (i=n-1;i>=0;i--) //从后往前比较，1分**

**if(a[i]>k) /若K较小，元素后移，2分**

**a[i+1]=a[i];**

**else //若K不小，插入，2分**

**{ a[i+1]=k;**

**break; //结束循环，1分**

**}**

**if(i<0) //若比最小还小，插到最前面，2分**

**a[0]=k;**

**for (i=0;i<=n;i++) //显示插入后数组，1分**

**printf("%d,",a[i]);**

**return 0;**

**}**

**2．（14分）**

**1）**

**struct BorrowedRecord //共2分**

**{**

**int no;**

**Char bname[32];**

**Date bdate;**

**}; //错一处扣1分，扣完为止**

**2）**

**void Sort(struct BorrowedRecord brecs[], int N) //共4分**

**{**

**int i,j;**

**struct BorroredRecord tbi;**

**for(i=0;i<N-1;i++)**

**for(j=0;j<N-i-1;j++) //循环都正确得1分**

**{**

**if(binfo[j].bdate.year>binfo[j].bdate.year //1分**

**||Binfo[j].bdate.year == binfo[j].bdate.year**

**&& binfo[j].bdate.month>binfo[j].bdate.bmonth**

**|| Binfo[j].bdate.year == binfo[j].bdate.year**

**&& binfo[j].bdate.month==binfo[j].bdate.month**

**&& binfo[j].bdate.day>binfo[j].bdate.day)**

**{**

**//正确完成交换得2分，错一处扣1分，扣完为止**

**tbi = brecs[j];**

**brecs [j] = brecs [j+1];**

**brecs [j+1]=tbi;**

**}**

**}**

**}**

**3）**

**void QueryExpire(struct BorrowedRecord brecs[], int N, Date qdate)**

**{//共计8分**

**int mdays[12]={31,28,31,30,31,30,31,31,30,31,30,31}; //1分**

**int i,j,year,month,day,count=0;**

**for(i=0; i<N; i++)**

**{**

**//将当前借阅记录的借阅时间增加60天，共计5分**

**year = brecs[i].bdate.year;**

**month = brecs[i].bdate.month;**

**day = brecs[i].bdate.day;**

**day += 60; //1分**

**if(year%400==0 || year%100!=0&&year%4==0) //1分**

**mdays[1]=29；**

**else**

**mdays[1]=28;**

**while(day>mdays[month]) //1分**

**{**

**day = day-mdays[month]; //1分**

**month++;**

**if(month==13)**

**{**

**month=1; //month和year的赋值1分，**

**year++; //与上面的if语句一起得分**

**if(year%400==0||year%100!=0&&year%4==0)**

**mdays[1]=29；**

**else**

**mdays[1]=28;**

**}**

**}**

**if(year<qdate.year**

**|| year==qdate.year && month<qdate.month**

**|| year==qdate.year && month ==qdate.month**

**&& day<qdate.month ) //1分**

**{**

**printf(“%d %d %s\n”**

**, ++count, brecs[i].no,brecs[i].bname); //1分**

**}**

**}**

**}**

**3．（14分）**

**//(1) 共2分**

**struct N {**

**int d; //1分**

**struct N \*next; //next加指针，1分**

**};**

**//(2) 共4分**

**int sym(struct N\* head)**

**{ struct N \*p=head;**

**int a[1000],n=0;**

**while (p != NULL) //链表导入数组2分**

**{ a[n++]=p->d; p=p->next;**

**}**

**//对称性判断，直接在数组a上进行，2分**

**for(int i=0,j=n-1;i<j;i++,j--)**

**if (a[i]!=a[j]) return 0;**

**return 1;**

**}**

**//(3) 共8分**

**//返回值，1分，形式参数，1分**

**struct N\* left(struct N\* A)**

**{ struct N\* B = NULL, \*temp1, \*temp;**

**struct N\* p = A;**

**int n=0;**

**while (p != NULL) //求串长1分**

**{ n++;**

**p=p->next;**

**}**

**p=A;**

**for(int i=0;i<(n+1)/2;i++) //循环左边链表1分**

**{ temp = (struct N\*)malloc(sizeof(struct N)); //malloc 1分，sizeof 1分，强制转换不计分**

**temp->d = p->d;**

**temp->next = NULL;**

**//插入结点逻辑，2分**

**if (B == NULL)**

**B = temp;**

**else**

**temp1->next = temp;**

**temp1 = temp;**

**p = p->next;**

**}**

**return B;**

**}**