电子科大*计算机学院*2012届保研复试—笔试部分

(回忆版,无答案)

考试时间:2011年10月14日8:30-10:30

(总分:100)

1. 程序改错:说明程序的错误原因并改正.(共20分)
2. 下面程序段的功能是交换两个字符数组的内容(每个字符串字符数均不超过100).(8分)

void StrSwap(char \*pa,char \*pb)

{ char \*temp;

temp = pa;

pa = pb;

pb = temp;

}

该程序中pa,pb两个字符指针变量只是形式参数，刚开始pa指向数组pa[100]的的首元素，指向pb[100]首元素；交换后pa指向数组pb[100]的的首元素，指向pa[100]首元素；而且该程序段运行结束后pa,pb两个变量由内存回收，因此最后两个字符数组的内容没有改变。

改正：

void copy(char \*pa,char \*pb)

{

while(\*pb!=0)

{

\*pa=\*pb;

pa++;

pb++;

}

\*pa=0;

}

void StrSwap(char \*pa,char \*pb)

{

char temp[100];

copy(temp,pa);

copy(pa,pb);

copy(pb,temp);

}

1. 程序段如下.(12分)

char a[] = “House”;

char \*b = “House”;

//b[2] = ‘r’; //不能对字符常量赋值

a[2] = ‘r’;

b = a;

b[2] = ‘r’;

//a = b; a是一个地址常量，不能把地址赋给一个常量

1. 简答题
2. 下面程序中，arr[]、numb、item分别是整型数组、数组元素个数、某一整数，程序功能是遍历数组arr，查找与item相等的元素，并输出该元素的下标。但是此程序不严谨，请问它可能导致什么样的异常结果？为什么？(5分)

for(numb = 0;arr[numb] != item;numb --);

printf(“%d”,numb);

// 只能遍历首元素，之后数组下标会越界，而且最多只能找到一个。

可以改为：

for( ; numb>0 ; numb--)

if( arr[numb-1] == 3)

printf("%d ",numb-1);

1. 下面程序段中各个常量、变量分别存储在内存中的什么位置、各按什么样的顺序存储、各占多少个字节？(提示：整型变量占2个字节，字符占1个字节，指针占4个字节)(10分)

int k; //全局未初始化区

void main() //栈区(stack),存放函数的参数值，局部变量的值等

{

char \*p = “hello”; //hello\0 在常量区, p在栈区

char q[]= ”hello”; //hello 在常量区, q在栈区

char ch; // 栈

int k; // 栈

func(k);

}

void func(int m) // 栈

{

int n; // 栈

……

}

1. 在调用函数时,如果形参和实参分别是下列情况,则相应的调用方式是什么?(5分)
2. 实参和形参都是数组元素//传值调用（值传递）
3. 形参是指针//传地址调用（地址传递）
4. 实参和实参都是数组//传地址调用（地址传递）
5. 编程题(共60分)
6. 编写一个函数，使之能完成以下功能：把一个字符串逆序排列。(10分)

方法一：

#include "stdio.h"

#include "string.h"

void reverse(char str[])

{

char ch;

int i,l=strlen(str);

for(i=0;i<l/2;i++){

ch=str[i];

str[i]=str[l-i-1];

str[l-i-1]=ch;

}

}

int main()

{

char ss[]="abcdefg";

reverse(ss);

printf("%s",ss);

return 0;

}

方法二：

# include <stdio.h>

# include <string.h>

int change(char \*str)

{

int i,len=0;

char c;

for( ; str[len]!='\0'; len++); //求字符串长度

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

{

c=str[i];

str[i]=str[len-i-1];

str[len-i-1]=c;

}

return len;

}

void main()

{

int i,len;

char str[20];

printf("请输入一个字符串：");

gets(str);

len=change(str);

printf("将字符串中的字符互换输出为：");

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

{

printf("%c",str[i]);

}

printf("\n");

}

//

#include<stdio.h>

void reverse(char \*s)

{

int len=strlen(s);

int i;

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

{

char t=s[i];

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

s[len-i-1]=t;

}

}

void main()

{

char s[]="uoy evol i";

reverse(s);

puts(s);

}

1. 编写一个函数，使之能完成以下功能：利用递归方法找出一个数组中的最大值和最小值，要求递归调用函数的格式如下：

MinMaxValue(arr,n,&max,&min)，其中arr是给定的数组，n是数组的个数，max、min分别是最大值和最小值。(15分)

#include "stdio.h"

#define N 10

void MinMaxValue(int arr[],int n,int \*max,int \*min)

{

if(n>=0)

{

if(\*max<arr[n])

\*max=arr[n];

if(\*min>arr[n])

\*min=arr[n];

MinMaxValue(arr,n-1,max,min);

}

}

int main()

{

int max=-32768,min=32767;

int a[]={1,54,23,65,87,12,54,87,98,233};

MinMaxValue(a,N-1,&max,&min);

printf("Max=%d,Min=%d\n",max,min);

return 0;

}

方法二：

#include "stdio.h"

#define N 10

void MinMaxValue(int arr[],int n,int \*max,int \*min)

{

if(n>0)

{

if(\*max<arr[n])

\*max=arr[n];

if(\*min>arr[n])

\*min=arr[n];

MinMaxValue(arr,n-1,max,min);

}

}

int main()

{

int a[]={1,-8,23,65,87,12,54,887,98,233};

int max=a[0],min=a[0];

MinMaxValue(a,N-1,&max,&min);

printf("Max=%d,Min=%d\n",max,min);

return 0;

}

//

#include<stdio.h>

void MinMaxValue(int arr[],int n,int \*max,int \*min)

{

if(n==0) return;

if(arr[n-1]>\*max) \*max=arr[n-1];

if(arr[n-1]<\*min) \*min=arr[n-1];

MinMaxValue(arr,n-1,max,min);

}

void main()

{

int arr[]={1,3,4,2,56,5};

int max=arr[0],min=arr[0];

MinMaxValue(arr,sizeof(arr)/sizeof(int),&max,&min);

printf("%d-%d\n",max,min);

}

1. 编写一个函数，使之能完成以下功能：把file1.doc的内容全部复制到file2.doc中，file1.doc中全部是字符(含空格)，要求复制时，在file2.doc中的每一行都要加上行号，例如：行号\*(其中“\*”表示具体的数字)。最后该函数返回file1.doc中的字符个数(不包括空格)。(10分)

方法一：

#include "stdio.h"

int exam()

{

char ch;

int count=0,row=0;

FILE \*fp1,\*fp2;

fp1=fopen("file1.doc","r+");

while(!feof(fp1))

{

ch=getc(fp1);

fp2=fopen("file2.doc ","a+");

if(ch!=' ')

count++;

if(ch=='\n')

{

row++;

fprintf(fp2,"行号:%d",row);

fprintf(fp2,"\n");

}

else

{

putc(ch,fp2);

}

fclose(fp2);

}

fclose(fp1);

return count;

}

int main()

{

int sum=exam();

printf("Sum=%d\n",sum);

return 0;

}

方法二：

# include <stdio.h>

# include <stdlib.h>

void main()

{

FILE \*fp1,\*fp2;

int count=0, row=0;

char c;

fp1=fopen("file1.txt","r"); //file 在当前 源程序目录下, 文本文件可以在运行后看到结果

fp2=fopen("file2.txt","w");

if(fp1== NULL)

{

printf("文件不能打开!\n"); // 当前目录不存在 file1.txt 文件 或者 该文件损坏

exit(0);

}

if(fp2== NULL)

{

printf("文件不能打开!\n");

exit(0);

}

while( (c=fgetc(fp1)) != EOF )

{

if(c!=' ')

count++;

if(c=='\n')

{

row++;

fprintf(fp2,"行号：%d",row);

}

fputc(c,fp2);

}

fclose(fp1);

fclose(fp2);

printf("count=%d,row=%d\n",count,row);

}

//

#include<stdio.h>

void main()

{

FILE \*fin,\*fout;

int count=0,line=0;

if(!(fin=fopen("file1.txt","r")))

printf("error");

if(!(fout=fopen("file2.txt","w")))

printf("error");

fprintf(fout,"行号%d ",line++);

while(!feof(fin))

{

char ch=fgetc(fin);

putchar(ch);

if((ch!=' ')&& (ch!='\n'))

count++;

fputc(ch,fout);

if(ch=='\n')

fprintf(fout,"行号%d ",line++);

}

fprintf(fout,"总个数%d ",count--);

printf("总个数%d ",count--);

fclose(fin);

fclose(fout);

}

1. 编写一个完整的程序，使之能完成以下功能：从键盘中输入若干个整数，用链表储存这些输入的数，并要求存储的顺序与输入的顺序相反。(10分)

方法一：

# include <stdio.h>

# include <malloc.h>

struct node

{

int val;

struct node \*next;

};

//创建链表

struct node\* createList()

{

struct node \*head,\*p;

int i,len,val;

//建立头结点

head = (struct node\*)malloc( sizeof(struct node) );

if(head==NULL)

{

printf("空间申请失败!\n");

return NULL;

}

head->val=NULL;

head->next=NULL;

printf("请输入结点数目：\n");

scanf("%d",&len);

printf("\n\n");

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

{

p = (struct node\*)malloc( sizeof(struct node) );

if(p==NULL)

{

printf("空间申请失败!\n");

return NULL;

}

printf("请输入第%d个结点的值：\n", i+1);

scanf("%d",&val);

p->val=val;

p->next = head->next;

head->next=p;

}

return head;

}

void traverse(struct node \*head)

{

struct node \*p=head->next;

if(p == NULL)

{

printf("链表为空!\n");

return ;

}

while(p!=NULL)

{

printf("%d ",p->val);

p=p->next;

}

printf("\n");

}

void main()

{

struct node \*head;

head = createList();

printf("\n链表创建完成并遍历输出:\n\n");

traverse(head);

}

方法二：

#include "stdio.h"

#include "stdlib.h"

struct sList

{

int data;

struct sList \*next;

};

struct sList \*head=NULL;

struct sList \*creat() //创建链表

{

int n;

struct sList \*p,\*q;//q new

head=(struct sList \*)malloc(sizeof(struct sList));

head->next=NULL;

p=head;

while(1)

{

printf("(0结束)n=");

scanf("%d",&n);

if(n<=0) break;

else{

q=(struct sList \*)malloc(sizeof(struct sList));

q->data=n;

p->next=q;

q->next=NULL;

p=q;

}

}

return head;

}

void ShowList(struct sList \*head) //显示链表

{

struct sList \*p=head->next;

while(p!=NULL){

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

p=p->next;

}

}

int main()

{

struct sList \*p,\*newhead,\*p1;

ShowList(creat());

printf("\n");

p=head->next;

newhead=head; //链表倒置

newhead->next=NULL;

while(p!=NULL)

{

p1=p;

p=p->next;

p1->next=newhead->next;

newhead->next=p1;

}

head=newhead;

printf("\n");

ShowList(head);

return 0;

}

//

#include<stdio.h>

#include<malloc.h>

struct Node

{

struct Node\*next;

int data;

};

void main()

{

int i=0,n;

struct Node \*head,\*p;

printf("格式:第一行数据的个数，第二行，输入数据空格隔开\n");

scanf("%d",&n);

head=malloc(sizeof(struct Node));

head->next=NULL;

for(;i<n;i++)

{

int temp;

scanf("%d",&temp);

p=malloc(sizeof(struct Node));

p->data=temp;

p->next=head->next;

head->next=p;

}

p=head->next;

while(p)

{

printf("%d\n",p->data);

p=p->next;

}

}

1. 编写一个完整的程序，使之能完成以下功能：一段名为file.c的程序，该程序中含有括号，现要检查程序中的括号是否配对，提示：利用堆栈实现。(15分)

#include <stdio.h>

#define Stack char

#define Max 999

int judge(char p[])

{

int len=strlen(p);

int top=-1,i;

Stack s[Max];

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

{

switch(p[i])

{

case '(':

case '[':

case '{':

s[++top]=p[i];

break;

case ')':

if(s[top]=='(')

top--;

else return 0;

break;

case ']':

if(s[top]=='[')

top--;

else return 0;

break;

case '}':

if(s[top]=='{')

top--;

else return 0;

break;

}

}

if(top==-1)

return 1;

else

return 0;

}

void main()

{

char p[Max];

FILE \* fin;

if(!(fin=fopen("file.txt","r")))

printf("failed");

while(!feof(fin))

{

fscanf(fin,"%s",p);

}

printf("%d\n",judge(p));

fclose(fin);

}

**附加：(第二批笔试变化题目)**

1.将输入的一个数质因分解，如Input 90，Print:90=2\*3\*3\*5.

方法一：

#include <stdio.h>

#include <math.h>

int a[100];

int isPrime(int n)

{

int i;

for(i=2;i<=sqrt(n);i++)

if(n%i==0)

return 0;

return 1;

}

int main()

{

int i,j,in;

while(scanf("%d",&in)&&in>-1)

{

if(in==0||in==1)

printf("没有质因数");

else

{

j=0;

printf("%d=",in);

while(in>3)

{

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

{

if(isPrime(i)&&in%i==0)

break;

}

in=in/i;

a[j++]=i;

}

if(in>1)

a[j++]=in;

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

{

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

if(i<j-1)

printf("\*");

}

printf("\n");

}

}

return 0;

}

方法二

# include <stdio.h>

void main()

{

int i,j,k,val;

printf("请输入一个正整数：");

scanf("%d",&val);

printf("将该正整数分解质因数输出为： ");

printf("%d=",val);

//以下为算法

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

{

while(val!=i)

{

if(val%i == 0)

{

printf("%d\*",i);

val=val/i;

}

else

break ;

}

}

printf("%d",val);

printf("\n");

}

2.编程求，x,n键盘输入。

方法一：

# include <stdio.h>

void main()

{

float sum=1, sum1=1, sum2=1, sum3, x;

int i, n, flag=-1;

printf("请输入x的值：\n");

scanf("%f", &x);

printf("请输入n的值：\n");

scanf("%d", &n);

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

{

sum1 = sum1\*x;

sum2 = sum2\*i;

sum3 = sum1/sum2 \* flag;

flag = -flag;

sum = sum + sum3;

}

printf("Sum=%f\n", sum );

}

方法二

#include "stdio.h"

#include "math.h"

int nJie(int a);

int main()

{

int i,n,sign;

float ex=1,temp,x,fenmu,fenzi;

printf("n=");

scanf("%d",&n);

printf("x=");

scanf("%f",&x);

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

fenzi=pow(x,i);

fenmu=nJie(i);

temp=fenzi/fenmu;

if(i%2==0) sign=1;

else sign=-1;

ex+=sign\*temp;

}

printf("ex=%6.2f",ex);

return 0;

}

int nJie(int a)

{

int i,ex=1;

for(i=1;i<=a;i++)

ex\*=i;

return ex;

}