**浙江大学2010–2011学年冬学期**

**《程序设计基础及实验》课程期末考试试卷**

课程号： 21186020\_\_，开课学院：\_计算机学院\_\_

考试试卷：√A卷、B卷（请在选定项上打√）

考试形式：√闭、开卷（请在选定项上打√），允许带\_\_\_\_∕\_\_入场

考试日期： 2011 年 01 月 19 日,考试时间： 120 分钟

**诚信考试，沉着应考，杜绝违纪。**

**考生姓名： 学号： 所属院系： \_**

**(注意：答题内容必须写在答题卷上，写在本试题卷上无效)**

试题一、单选题（每小题2分，共20分）

1. 表达式 (4>3>2) 的值为\_\_\_\_\_\_\_.

|  |  |  |  |
| --- | --- | --- | --- |
| A．0 | B．1 | C．2 | D．表达式错误 |

1. 下面哪个表达式符合C语言语法\_\_\_\_\_\_\_.

|  |  |  |  |
| --- | --- | --- | --- |
| A．0.5%2 | B．3&(0.5+0.5) | C．4.6<<2 | D．2==(3/2) |

1. 若有定义：int a[2][3]; 以下选项中能对a数组元素正确引用的是\_\_\_\_\_\_\_.

A．a[2][!1] B．a[2][3] C．a[0][3] D．a[1>2][ !1]

1. 以下程序段完全正确的是\_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| A．int \*p;  scanf（"%d", ＆p）; | B．int \*p;  scanf（“%d”, p）; |
| C．int k, \*p=&k;  scanf（"%d", p）; | D． int k, \*p;  \*p= &k;   scanf（“%d”, p）; |

1. 有定义语句：int \*p[4];以下选项中与此语句等价的是\_\_\_\_\_\_\_.

|  |  |  |  |
| --- | --- | --- | --- |
| A．int  p[4]; | B．int  \*\*p; | C．int  \*（p[ 4 ]）; | D．int （\*p）[ 4 ]; |

1. 若要定义一个具有4个元素的整型数组，以下错误的定义语句是

|  |  |  |  |
| --- | --- | --- | --- |
| A．int  a[4]={0}; | B．int b[]={0, 0, 0, 0}; | C．int c[1+3]; | D．int  i=4, d[ i ]; |

1. 若有定义：static int a［3］［4］= {0}; 则下面正确的叙述是\_\_\_\_\_\_\_.

|  |
| --- |
| A. 只有元素a［0］［0］可得到初值0 |
| B. 此说明语句不正确 |
| C. 数组a中各元素都可得到初值，但其值不一定为0 |
| D. 数组a中每个元素均可得到初值0 |

1. 对于以下变量定义，正确的赋值是\_\_\_\_\_\_\_.

int \*p[3], a[3];

|  |  |
| --- | --- |
| A．p=a | B．p=&a[0] |
| C．\*p=a | D．p[0]=\*a |

1. 结构和变量定义如下，则对于“\*p->str++”表达式语句，下面叙述正确的是\_\_\_\_\_\_\_.

struct { int no; char \*str;} a={1,”abc”}, \*p=&a;

|  |  |
| --- | --- |
| A．\* 作用在指针p上 | B．++ 作用在指针str上 |
| C．++ 作用在str所指的内容上 | D．表达式语句有错 |

1. 对于下面变量定义，赋值使用方法不正确的是\_\_\_\_\_。

struct Student { long num; char name[20]; } st1, st2={101,”Tom”}, \*p=&st1;

|  |  |
| --- | --- |
| A．st1=st2 | B．p->name=st2.name |
| C．p->num=st2.num | D．\*p=st2 |

试题二、填空题（每小题2分，共30分）

1. 下面程序段的输出结果是\_\_\_\_\_\_\_.

printf(“%d”,( 5 << 1 ) & ( 0x1f ^ 31 ));

1. 若’a’的ASCII码值为97，那么下面程序段的输出是\_\_\_\_\_\_\_.

printf(“%d”,’d’+027);

1. 下面代码段输出结果是\_\_\_\_\_\_\_.

int i,j,k;

i=j=k=1;

k+=1+(i++)+(++j);

printf("%d#%d#%d",i,j,k);

1. 下面代码段输出结果是\_\_\_\_\_\_\_.

int s=-1;

printf(“%d”,(unsigned int)s );

1. 表达式 1<<2+3的值为\_\_\_\_\_\_\_.
2. 执行下面程序代码后，s值为\_\_\_\_\_\_\_.

int i, s;

for (i=10,s=0; ;i--) {

if (i==0) break;

else if(i%2) continue;

s+=i;

}

1. 执行下面程序代码后，s值为\_\_\_\_\_\_\_.

int a=1, b=2, s=0;

switch (a>b) {

default: switch(s) {

case 0:s+=1;

default:s+=2;break;

}

case 1: s+=3; break;

}

1. 以下程序段的输出结果是\_\_\_\_\_\_\_.

#define MOD(x, y) x%y

int a = 15, b = 100, z;

z = MOD(b, a);

printf("%d", z++);

1. 对于以下递归函数f，调用函数f(2, '1', '2', '3')的输出结果是\_\_\_\_\_\_\_.

void f(int n, char a, char b, char c)

{ if(n == 1)

printf("%c%c#", a, b) ;

else{

f(n-1, a, c, b);

printf("%c%c#", a, b);

f(n-1, c, b, a);

}

}

1. 以下程序的输出结果是\_\_\_\_\_\_\_.

#include <stdio.h>

void f( int \*a, int b )

{

static int k=0;

\*a += ++k;

b +=2;

}

void main()

{ int x=2, y=4;

f(&x, y);

f(&y, x);

printf("%d#%d", x, y);

}

1. 以下程序的输出结果是\_\_\_\_\_\_\_.

#include <stdio.h>

void f( int \*a, int \*b )

{

int \*t;

t = a, a = b, b = t;

}

void main()

{ int x=1, y=2;

f(&x, &y);

printf("%d#%d", x, y);

}

1. 执行命令行程序：find –nx ould text.txt时，\*++argv[1]的值是\_\_\_\_\_\_\_.
2. 下列程序输出\_\_\_\_\_\_\_.

#include <stdio.h>

#include <string.h>

void main( )

{ char x[ ]=”STRING”;

x[0] = 0；x[1] = ’\0’；x[2] = ’0’;

printf(”%d#%d”，sizeof(x)，strlen(x));

}

1. 用 typedef 定义新类型名PNODE，使得PNODE表示含有10个struct Node结构指针型元素的数组类型\_\_\_\_\_\_\_.
2. 下面程序段的输出是\_\_\_\_\_。

char \*a[2]={"one", "two"}, \*\*p=a;

printf("%s#", \*(p++)+1);

printf("%c", \*\*p-1);

试题三、程序阅读题（每小题5分，共30分）

1. 如果指针的长度为4字节，则下列程序的输出是\_\_\_\_\_\_\_.

#include <stdio.h>

short x[16][32], \*y[32], i, j;

int main(void)

{

while(i < sizeof(x)/sizeof(x[0])) {

y[i]=x[i];

for(j=0; j < sizeof(y); j++) {

x[i][j/sizeof(y[0])] = j + i \* sizeof(x[0]);

}

i++;

}

for(i=0;i<5;i++) printf("%d#", y[1][i]);

return 0;

}

1. 下列程序的输出是\_\_\_\_\_\_\_.

#include <stdio.h>

char s1[80]="abc56cdefghi\0jklm";

int main(void)

{

int j=0, k=0, c;

char ch, bf=0, s2[80];

while(ch=s1[j++]) {

if(ch - bf != 1) {

s2[k++] = ch;

c=0;

} else if(s1[j] - ch != 1) {

if (c) s2[k++] = '-';

s2[k++] = ch;

} else

c++;

bf = ch;

}

s2[k]='\0';

printf("%s", s2);

return 0;

}

1. 当输入：a1b2c3d<回车>时，下列程序的输出是\_\_\_\_\_\_\_.

#include <stdio.h>

void main()

{

char \*p,s[8];

long n;

p=s;

gets(p);

while (\*p !='\0'&& (\*p>'8'||\*p<'0')) p++;

n=\*p-'0';

while(\*++p !='\0')

if (\*p>'8'||\*p<'0') continue;

else n=n\*8+\*p-'0';

printf("%ld",n);

}

1. 下列程序运行时输出结果为\_\_\_\_\_\_\_.

#include <stdio.h>

int findStr (char \*str, char \*substr)

{ int n;

char \*p , \*r;

n=0;

while ( \*str ){

p=str;

r=substr;

while(\*r)

if(\*r==\*p) { r++; p++; }

else break;

if(\*r=='\0')

n++;

str++;

}

return n;

}

void main()

{

char s1[80]="AAscA\\101\101scA\nAscA00AscA\0AscAs", s2[10]="Asc";

printf("%d#", findStr(s1,s2));

printf("%d#", findStr(s1+10,s2));

printf("%d#\n", findStr(s1+25,s2));

}

1. 在调用执行下列showProcess函数时，如果第一个printList()函数调用输出的结果是：4 6 2 5 1 8，则第二个printList()函数调用输出的结果应该是\_\_\_\_\_\_\_.

struct node {

int data;

struct node \*next;

};

struct node \*process(struct node \*h, int n, int m)

{ struct node \*p, \*q;

p=h;

while (p!=NULL)

if (p->data >=n && p->data <=m) {

if (p==h) { p=p->next; free(h); h=p; };

else { q->next=p->next;

free(p);

p=q->next;

}

} else {

q=p;

p=p->next;

}

return h;

}

void printList(struct node \*h)

{ struct node \*p=h;

while (p!=NULL) {

printf(“%d “, p->data);

p=p->next;

}

}

void showProcess(struct node \*h)

{ printList(h); //第一个printList

h=process(h,4,6);

printList(h); //第二个printList

}

1. 下列源程序经编译链接后的可执行程序为prog.exe，以命令行方式运行：prog file1 file2 ，其中file1、 file2分别是两个文本文件。假如file1中的内容为aaabbcdddd，则程序运行后file2中的内容为\_\_\_\_\_\_\_.

#include <stdio.h>

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

{ FILE \*fp1,\*fp2;

char c, last= -1;

int count=0;

if ((fp1=fopen(\*++argv,"r"))==NULL) exit(0);

if ((fp2=fopen(argv[1],"w"))==NULL) exit(0);

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

count++;

if (c!=last) {

if (last!=-1) fprintf(fp2,"%d%c", count, last);

last=c;

count=0;

}

}

if (count!=0) fprintf(fp2,"%d%c", count, last);

fclose(fp1);

fclose(fp2);

}

试题四、程序（段）填空题（每空2分，共20分）

1. 下面带命令行参数的程序calculator.c能够完成正整数的加法或者减法的计算。该程序从命令行获得加法或者减法的指示及相应的操作数。例如，

命令行输入> ***calculator 213 312 +***

程序输出为> **The result is 525**

命令行输入> ***calculator 21 1 –***

程序输出为> **The result is 20**

其他形式的命令行输入如命令行参数个数不符或操作非法等将输出**Invalid operation!**。

/\* command-line program： calculator.c \*/

#include <stdio.h>

#include <string.h>

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

{

int result;

int strTOint(char \*str);

if(　　 **(1)**　 　) {

printf(""Invalid operation!");

return -1;

}

switch(　　 **(2)**　 　) {

case '+':result = 　　 **(3)**　 　; break;

case '-': result = 　　 **(4)**　 　; break;

default: printf("Invalid operation!"); return –1;

}

printf("The result is %d",result);

return 0;

}

int strTOint(char \*str)

{

int i, length;

int sum = 0;

length = strlen(str);

for(i = 0;i < length; ++i,++str)

**(5)**　 　;

return sum;

}

1. 有两个包含学生成绩的单向链表h1和h2，其中的结点顺序都是按成绩从高到低排列，结点中含有学生姓名(字符串方式)和成绩（整型）信息。下列函数merge将这两个链表中的信息按成绩从高到低顺序合并输出到out.dat文本文件中。要求，输出文件中每个学生信息占文本文件的一行：姓名占20个字符位（如果学生姓名超过20个字符只取前20，左对齐），成绩占3位（右对齐）。

struct node {

char name[40]；

int score;

struct node \*next;

};

printStudentInfo(FILE \*fp, struct node \*p)

{

fprintf(　　 **(6)**　 　);

}

void merge(struct node \*h1, struct node \*h2)

{

FILE \*fp;

struct node \*p, \*q;

p=h1;

q=h2;

if ((　　 **(7)**　 　)==NULL) exit(0);

while(p!=NULL && q!=NULL)

if (　　 **(8)**　 　) {

printfStudentInfo(fp, p);

p=p->next;

} else {

printfStudentInfo(fp, q);

q=q->next;

}

while (　　 **(9)**　 　) {

printfStudentInfo(fp, p);

p=p->next;

}

while (q!=NULL) {

printfStudentInfo(fp, q);

**(10)**　 　;

}

fclose(fp1);

fclose(fp2);

return;

}