**诚信考试 沉着应考 杜绝违纪**

**浙江大学20\_08 –20 09 学年 冬 学期**

**《程序设计基础及实验》课程期末考试试卷（B）**

**开课学院： 计算机学院 ，考试形式： 闭 卷，允许带 / 入场**

**考试时间： 2009 年 1月 15 日,所需时间： 120 分钟，任课教师 \_\_\_\_\_\_\_\_**

**考生姓名： 学号： 专业：**

**(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. 若变量已正确定义，执行“scanf("%d%c%lf=%lf", &op1, &op, &op2, &res);”，输入 后，op1的值为3，op的值为’+’，op2的值为0.1，res的值为1.2。

|  |  |  |  |
| --- | --- | --- | --- |
| A．3 +0.1=1.2 | B．3+1E-1=1.2 | C．3,+,0.1=1.2 | D．3+0.1 1.2 |

1. 程序段\_\_\_\_\_与语句 k=a>b?(b>c?1:0):0; 功能等价。

|  |  |
| --- | --- |
| A．if((a>b)&&(b>c)) k=1;  else k=0; | B．if((a>b)||(b>c))k=1;  else k=0; |
| C．if（a>b）k=1;  else if（b>c）k=1;  else k=0; | D．if(a<=b)k=0;  else if(b>c) k=1; |

1. 对下面程序描述正确的是\_\_\_\_\_。

# include <stdio.h>

main()

{ int i;

for(i=0;--i;i--)

printf("%d ",i);

}

|  |  |
| --- | --- |
| A．编译通不过 | B．printf被执行0次 |
| C．printf被执行有限次 | D．是无限循环，俗称死循环 |

1. 执行完下列语句后，i,j值分别为\_\_\_\_\_。

int i=5, j=5;

if (i=j==5) j=3; else j=4;

|  |  |  |  |
| --- | --- | --- | --- |
| A．i=5 j=3 | B．i=5 j=4 | C．i=1 j=3 | D．i=1 j=4 |

1. 计算x的符号函数值的条件表达式正确的是\_\_\_\_\_。

（说明：x大于0，x的符号函数值是1；x等于0，是0；x小于0，是-1。）

|  |  |
| --- | --- |
| A．x>0?1:x=0?0:-1 | B．x>=0?1:x<0?-1:0 |
| C．(x>0?1:x==0)?0:-1 | D．x>0?1:x==0?0:-1 |

1. C语言中可以使用关键字static来定义静态变量。现要求定义一个静态的整形变量，并将其初始化为-1，下列定义正确的是\_\_\_\_\_\_\_\_。

|  |  |
| --- | --- |
| A．static int foo = -1; | B．int foo static = -1; |
| C．int foo = static -1; | D．int foo = -1 static; |

1. 下列程序段输出\_\_\_\_\_\_\_\_。

char c[20]=“It\’s\timetogo.\n”;

printf(“%d”,strlen(c));

|  |  |  |  |
| --- | --- | --- | --- |
| A．14 | B．15 | C．19 | D．20 |

1. struct stud {

int num;

char name[20];

float score;

} a[3], x,\*p;

\_\_\_\_\_\_\_\_是错误的。

|  |  |
| --- | --- |
| A．p=a+2; | B．p=(struct stud \*)malloc(sizeof(x)); |
| C．p=(struct stud \*)malloc(a); | D．p=(struct stud \*)malloc(sizeof(struct stud)); |

1. 若有如下定义：

#define PI 3.14

float \*p,m=PI;

则\_\_\_\_\_是正确的赋值语句。

|  |  |  |  |
| --- | --- | --- | --- |
| A．p=&m; | B．p=m; | C．\*p=m; | D．\*p=&m; |

1. fp=fopen(\_\_\_\_\_\_,”w”);

上述fopen语句中正确的参数是 \_\_\_\_\_。

|  |  |
| --- | --- |
| A．text.txt | B．c:\user\text.txt |
| C．”c:\user\text.txt” | D．”c:\\user\\text.txt” |

试题二、填空题（每小题2分，共20分）

1. 若定义：int x, y; 则表达式x=(y=1)/2&&(x=1)的值是\_\_\_\_\_。
2. 若变量已正确定义并赋值，则表达式(0<x<5)的值为\_\_\_\_\_。
3. 表达式int(6.8)%4的值为\_\_\_\_\_。
4. 表达式(0xffff&010)的值为\_\_\_\_\_（十进制）。
5. 下面程序段的输出结果是\_\_\_\_\_。

int k, s;

for(k=1, s=0; ++k<5; k++){

if (k%2) break;

s += k;

}

printf("k=%d , s=%d", k, s);

1. 下面程序段的输出结果是\_\_\_\_\_。

int i,c[10]={6,5,4,3,2,1,0,-1,-2,-3}, \*pc;

for(i=0,pc=c; \*pc++&&i<10; i++,pc++) printf("%d#", \*pc);

1. 用for循环打印1 2 3 1 2 3 1 2 3，将下面程序段代码填写完整。

for (i=1; i<=9; i++) printf("%2d", \_\_\_\_\_);

1. 下列程序段执行后s值为\_\_\_\_\_。

int i=5, s=10;

do if (i%2) {continue;s-=i;} else s+=i; while (--i);

1. 假设下面程序的生成的可执行文件是prog.exe：

#include <stdio.h>

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

{ int i;

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

printf("%s%c", argv[i], (i == argc - 1 ? '\n': '#'));

return 0;

}

执行命令：prog.exe foo bar “foo bar”，得到的结果是\_\_\_\_\_\_\_\_\_\_\_\_\_。

1. 下面程序的输出结果是\_\_\_\_\_\_\_\_\_。

#include <stdio.h>

int \*foo(void)

{

static int bar = 123;

return &bar;

}

int main()

{

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

return 0;

}

1. 用 typedef 写出类型定义 ，使得AIP表示含有10个元素的整型指针数组类型。
2. 以下程序的运行结果是\_\_\_\_\_。

f (int \*x)

{

static int t=0;

t += \*x;

return t;

}

void main( )

{ int x=5,y=1;

f(&y);

printf("%d", f(&x));

}

1. 以下程序的运行结果是\_\_\_\_\_.

struct n{int x;char c;}

main() {

struct n a={10,’A’};

f(a);

printf(“%d#%c#”,a.x,a.c);

}

f(struct n b) {

b.x=20;b.c=’B’;

return 1;

}

1. 下列程序段的输出结果是\_\_\_\_\_.

struct st {

int x, y;

} a[4]={{10,20},{30,40},{50,60},{70,80}}, \*p=a;

printf("%d#", p++->x);

printf("%d#", ++(p->y));

printf("%d#\n", (a+3)->x);

1. 下列程序段的输出结果是\_\_\_\_\_.

char \*st[ ]={"HELLO","WORLD"};

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

试题三、程序阅读题（每小题6分，共30分）

1. 运行下列程序，当输入***Hello Everyone!<回车>*** 时，输出结果是\_\_\_\_\_\_\_。

#include <stdio.h>

#include <string.h>

#include <ctype.h>

void main( )

{

char s1[31],s2[31],s[61];

int s1n,s2n,n;

scanf("%s%s", s1,s2);

s1n=strlen(s1); s2n=strlen(s2);

n=0;

while(s[n]=s2[n++]);

while(1){

if(! (s[n]= tolower (s1[n-s2n]))) break ;

n++;

}

printf("%s\n",s);

}

1. 运行下列程序的输出结果是\_\_\_\_\_\_\_\_\_\_。

#include <stdio.h>

void fun( int n, int \*p )

{

int s;

if( n<=1 ) \*p=1;

else{

fun(n-1,&s);

\*p=n\*s;

}

}

void main()

{ int n;

fun(4, &n);

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

}

1. 下列程序的输出结果是\_\_\_\_\_\_\_\_\_\_。

#include <stdlib.h>

main()

{

int \*p[5], int \*ptr,i;

if((ptr=(int\*)malloc(100\*sizeof(int)))!=NULL)

{

for(i = 0; i < 100; i++) ptr[i] = 0;

p[0]=ptr;

for(i=1;i<5;i++)

p[i]=p[i-1]+7;

for(i=0;i<100;i++)

\*ptr++=i;

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

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

}

}

1. 下列程序的运行结果是\_\_\_\_\_\_\_\_\_\_。

#include <ctype.h>

#include <string.h>

void fun(char \*s)

{

int i,t;

char ts[81];

for(i=0,t=0;s[i]!=0;i+=2)

if(!isspace(\*s+i)&&(\*(s+i)!='B'))

ts[t++]=tolower(s[i]);

ts[t]='\0';

strcpy(s,ts);

}

int main()

{

char str[81]={"A B C D EF G"};

fun(str);

puts(str);

return 0;

}

1. 有一链表h如下图所示

h

8 

7

1

3

2

以下程序段执行后，输出结果是\_\_\_\_\_\_\_\_\_\_。

struct node {

int info;

struct node \*next;

} \*h, \*p, \*q, \*t;

q=NULL; p=h;

while (p!=NULL) {

t=p->next;

p->next=q;

q=p;

p=t;

}

h=q;

while(q!=NULL) {

printf("%d", q->info);

q=q->next;

}

1. 下列程序运行时输出结果为\_\_\_\_\_\_\_。

#include <stdio.h>

#include <math.h>

#define N 1000L

#define SQRTN ((long) sqrt(N))

int main(void)

{

int p[N + 1]; long n, s;

p[0] = 0; p[1] = 0;

for (n = 2; n <= N; n++)

p[n] = 1;

for (n = 2; n <= SQRTN; n++)

if (p[n])

for (s = 2; s <= (N / n); s++)

p[s \* n] = 0;

for (n = 10; n <= 30; n++)

if (p[n]) printf("%6ld", n);

printf("\n");

return (0);

}

试题四、程序（段）填空题（每空2分，共30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将输出**Invalid operation!**。

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

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

{

long result;

long strTOint(char \*str);

if(　　 **(1)** 　 　) {

printf("Invalid operation!"); return 0;

}

switch(　 **(2)**  　) {

case '+':

result = (　 　　**(3)** 　); break;

case '-':

result = (　 　**(4)** 　); break;

default:

printf("Invalid operation!");

}

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

}

long strTOint(char \*str)

{

int i, length;

long sum = 0;

length = strlen(str);

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

**(5)**　 　;

return sum;

}

1. 下列程序从一个文件in.txt中读入，并将文件的内容输出到out.txt中，要求将文件内容中‘{’‘}’之间的小写字母转换为大写(设‘{’‘}’均配对且不嵌套)。

例如，若in.txt文件内容为：

This is a test {just for C}

You are right!

则out.txt文件的内容将为：

This is a test {JUST FOR C}

You are right!

#include <stdio.h>

main()

{ 　　 **(6)**

char c;

int state=0;

if ((fp1=fopen("in.txt","r"))==NULL) exit(0);

if ((fp2=fopen("out.txt","w"))==NULL) exit(0);

while(　　 **(7)** 　 　) {

if (c=='{' || c== '}') state=　　**(8)** 　;

if (state==1 && c>='a' && c<='z') c= 　　 **(9)** 　 　;

**(10)**

}

fclose(fp1);

fclose(fp2);

}