1、利用递归函数f(n)，求解表达式f(n)=1-2+3-4+...+n\*(-1)^(n+1)的值，请完善下面的递归函数：

int f(int n)

{

if (n==1)

{

return 【1】;

}

if (n>1)

{

if (n%2==0)

{

return 【2】;

}

return 【3】;

}

return 0;

}

【1】处应填入的是：\_\_\_\_\_\_。

A) -1 B）1 C) 2 D）0

答案：B

【2】处应填入的是：\_\_\_\_\_\_。

A) f(n-1)-n B）f(n-1)+n C) f(n-1)+1 D）-n;

答案：A

【3】应填入的是：\_\_\_\_\_\_。

A) f(n-1)-n B）f(n-1)+n C) f(n-1)+1 D）n;

答案：B

2、要对全班100个同学的成绩,按照从高到低的顺序进行排序，以下程序是排序函数，其功能是将成绩数组score从大到小排序。请完善函数中的语句。

void sort(int scores[],int n )

{

int i,j,temp;

for(i=【4】;i<=n-1;i++)

for(j=0;j<【5】;j++)

if(【6】)

{

temp=scores[j];

scores[j]=scores[j+1] ;

scores[j+1]=temp;

}

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

printf("%d\n",scores[i]);

}

为保证程序循环次数最少，【4】处应填入：\_\_\_\_\_\_。

A) 0 B）1 C) -1 D）N/2

答案：B

为保证程序循环次数最少，【5】处应填入：\_\_\_\_\_\_。

A) N B）N-1 C) N-i-1 D）N-i;

答案：D

【6】处应填入：\_\_\_\_\_\_。

A) scores[j]<scores[j+1] B）scores[j]>scores[j+1]

C) scores[j]<scores[j-1] D）scores[j]>scores[j-1]

答案：A

3、根据程序的初始值，打印选中学生的姓名与年龄，补充部分程序，并选择程序的运行结果：

#include <stdio.h>

struct student{

int id;

char name[12];

int age;

};

void print(struct student \*p)

{

printf("name=%s\n",(\*p).name);/\*【7】\*/

printf("age=%d\n",【8】);/\*打印输出年龄\*/

}

main( )

{

struct student BIT[3]={{1001,"Zhang",19},

{1002,"Zeng",20},

{1003,"Zang",21},

};

struct student \*p=BIT+1;

print (p) ;

}

【7】处的输出结果为：\_\_\_\_\_\_。

A) Zhang B）Zeng C) Zang D）结果不确定

答案：B

【8】处输出该学生的年龄，应填入：\_\_\_\_\_\_。

A) p->age B）student.age C) BIT[1].age D）p.age

答案：C

1、以下程序完成 3 个操作：

（1）输入 10 个字符串（每串不多于 9 个字符），依次存放在数组 a 中，指针数组 str 中的

每个元素依次指向每个字符串的开始；

（2）输出每个字符串；

（3）从这些字符串中选出最小的那个串输出。

请从对应的一组选项中选择正确的选项填空。

#include <stdio.h>

#include <string.h>

main( )

{ char a[200] = { ’\0’ }, \*str[10], \*sp；

int i, k;

sp = 【1】 ;

for( i = 0；i < 10; i++ )

{ scanf( "%s", sp );

str[i] = sp;

k = strlen(sp);

sp += 【2】 ;

}

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

printf( "%s\n", str[i] );

k = 0;

for( i = 1; i < 10; i++ )

if( strcmp( str[i], str[k] ) 【3】 )

k = i;

printf( "%s\n", str[k] );

}

随米云打印 网址：sui.me

【1】A、’\0’ B、str C、a D、&a[1]

【2】A、1 B、k C、k + 1 D、k + i

【3】A、> 0 B、< 0 C、== 0 D、!= 0

2、在保存字符串时，对出现在连续位置的若干个字符，如这些字符在 ASCII 表中也是连续

出现，则可以将这些字符使用缩写形式进行保存，以节省保存空间。例如用 A-Z 表示

ABCD...XYZ，1-5 表示 12345，c-f 表示 cdef。以下程序是展开输出经过压缩保存的字符串。

以下程序仅处理大写英文字母、小写英文字母和数字的同类压缩，即对于形如 A-9，a-C 这

样的缩写不进行处理。请从对应的一组选项中选择正确的选项填空。

#include <stdio.h>

main( )

{ char sa[100];

int i = 0, k;

gets( sa );

do

{ putch( sa[i] );

i++;

if( sa[i] == ’-’ )

{ k = 1;

do

{ putch( 【4】 );

k++;

}while( 【5】 );

i++;

}

}while( sa[i] != ’\0’ );

printf( "\n" );

}

【4】A、sa[i-1] B、sa[i - 1] + k C、sa[i] D、sa[i] + k

【5】A、sa[i-1] + k != sa[i+1] B、sa[i] + k != sa[i+1]

C、sa[i-1] != sa[i+1] + k D、sa[i] != sa[i+1] + k

3、以下程序将文件 A.DAT 和 B.DAT 合并到文件 C.DAT 中，其中 A.DAT 和 B.DAT 为降序字符文

件，保存在当前路径下，文件结束后标志设为‘\0’。请从对应的一组选项中选择正确的选

项填空。

#include <stdio.h>

main( )

{ 【6】 \*in1, \*in2, \*out;

int flag1 = 1, flag2 = 1;

char a, b, c;

if( ( in1 = fopen("A.DAT", "r") ) == NULL )

{ printf( "Can’t open the file A.DAT!\n" );

exit( 1 );

}

if( ( in2 = fopen( "B.DAT", "r") ) == NULL )

{ printf( "Can't open the file B.DAT!\n" );

exit( 1 );

}

if( ( out = fopen( "C.DAT", 【7】 ) ) == NULL )

{ printf( "Can't open the file C.DAT!\n" );

exit( 1 );

}

while( !feof( in1 ) 【8】 !feof( in2 ) )

{ if( !feof( in1 ) && flag1 )

a = fgetc( in1 );

else

if( feof( in1 ) )

a = 【9】 ;

if( !feof( in2 ) && flag2 )

b = fgetc( in2 );

else

if( feof( in2 ) )

b = 【9】 ;

if( 【10】 )

{ c = a; flag1 = 1; flag2 = 0; }

else

{ c = b; flag1 = 0; flag2 = 1; }

fputc( c, out );

}

fclose( in1 );

fclose( in2 );

fclose( out );

}

【6】 A、char B、int C、float D、FILE

【7】 A、"r" B、"w" C、"a" D、"b"

【8】 A、|| B、&& C、== D、!=

【9】 A、0 B、1 C、’\n’ D、’\0’

【10】A、a < b B、a <= b C、a > b D、a >=