

① 当前作业

[» 21级第七次作业 \(图\)](#)[» 21级第六次作业 \(查找与排序\)](#)[» 21级第五次作业 \(树\)](#)[» 2021级 \(信息大类\) 数据结构综合作业 \(正确性和性能\)](#)[» 2021级 \(信息大类\) 数据结构综合作业 \(可扩展性\)](#)

② 历史作业

[» 21级第四次作业 \(栈和队\)](#)[» 21级第三次作业](#)[» 21级第二次作业](#)[» 21级第一次作业](#)

21级第一次作业 作业时间: 2022-03-03 18:00:00 至 2022-03-31 23:55:00

选择填空题: 20分, 编程题100分, 总分120分。

填写选择题的答案时, 可以用大写字母, 也可以用小写字母, 但不得用小括号。

作业时间:四周, 截止日期: 2022年3月31日晚上11点55分整。

总分: 120.00

-pencil 选择题

1. 首次提交时间:2022-03-03 19:04:50 最后一次提交时间:2022-03-03 19:06:47 得分: 1.00

已提交

下面程序段的运行结果是 D _____ 【正确答案: d】。

char a[7] = "abcdef", b[4] = "ABC";

strcpy(a, b);

printf("%c", a[1]);

(A) a

(B) A

(C) b

(D) B

2. 首次提交时间:2022-03-03 19:11:10 最后一次提交时间:2022-03-03 19:11:10 得分: 1.00

已提交

以下给字符数组str定义和赋值正确的是 B _____ 【正确答案: B】

(A) char str[10]; str = "China";

(B) char str[] = "China";

(C) char str[10]; strcpy(str, "abcdefghijklmn");

(D) char str[10] = "abcdefghijklmn";

3. 首次提交时间:2022-03-03 19:12:50 最后一次提交时间:2022-03-03 19:12:50 得分: 1.00

已提交

下面程序段的运行结果是 B _____ 【正确答案: B】。

char c[] = "\t\n\\\"0will\n";

printf("%d", strlen(c));

(A) 14

(B) 3

(C) 9

(D) 字符串有非法字符, 输出值不确定

4. 首次提交时间:2022-03-03 19:16:41 最后一次提交时间:2022-03-03 19:16:41 得分: 1.00

已提交

下述对C语言字符数组的描述中错误的是 C _____ 【正确答案: c】

(A) 字符数组可以存放字符串

(B) 字符数组的字符串可以整体输入,输出

(C) 可以在赋值语句中通过赋值运算符 "=" 对字符数组整体赋值

(D) 不可以用关系运算符对字符数组中的字符串进行比较

5. 首次提交时间:2022-03-03 19:20:20 最后一次提交时间:2022-03-03 19:20:20 得分: 1.00

已提交

下面能正确进行字符串赋值, 并且能确保字符串以'\0'结尾的操作是

C _____ 【正确答案: c】

(A) char s[5] = {"ABCDE"};

(B) char s[5] = {'A', 'B', 'C', 'D', 'E'};

(C) char *s; s = "ABCDE";

(D) char *s; scanf("%s", s);

6. 首次提交时间:2022-03-03 19:21:44 最后一次提交时间:2022-03-03 19:21:45

得分: 1.00

已提交

有下面的程序段:

```
char a[3], b[] = "China";
a = b;
printf("%s", a);
```

则 D 【正确答案: d】

- (A) 运行后将输出China
 (B) 运行后将输出Ch
 (C) 运行后将输出Chi
 (D) 编译出错

7. 首次提交时间:2022-03-03 19:22:27 最后一次提交时间:2022-03-03 19:22:27

得分: 1.00

已提交

阅读程序, 选择程序的运行结果:A _____

【正确答案: A】

```
#include <stdio.h>
int try(int );
int main()
{
    int x;
    x = try(5);
    printf("%d\n",x);
    return 0;
}
int try(int n)
{
    if(n>0)
        return ( n*try(n-2));
    else
        return (1);
}
```

- A.15
 B.120
 C.1
 D.前面的答案均不正确

8. 首次提交时间:2022-03-03 19:26:03 最后一次提交时间:2022-03-03 19:26:03

得分: 1.00

已提交

下面程序的功能是将字符串s中所有的字符“c”删除。请选择填空。

```
#include<stdio.h>
int main()
{
    char s[80];
    int i,j;
    gets(s);
    for(i=j=0;s[i]!='\0';i++)
        if(s[i]!='c') A _____
        s[j]='\0';
        puts(s);

    return 0;
}
```

- (A)s[j++]=s[i]
 (B)s[+ +j]=s[i]
 (C)s[j]=s[i];j++
 (D)s[j]=s[i]

9. 首次提交时间:2022-03-03 20:49:45 最后一次提交时间:2022-03-03 20:49:45 得分: 1.00

已提交

若有说明: char *language[] = {"FORTRAN", "BASIC", "PASCAL", "JAVA", "C"};则以下不正确的叙述是 D
【正确答案: d】

- (A) *(language+2) 表示字符串"PASCAL"的首地址
- (B) language[2]的值是字母P
- (C) language是一个字符型指针数组,它包含5个元素,每个元素都是一个指向字符串的指针
- (D) language包含5个相同长度的数组

10. 首次提交时间:2022-03-03 19:27:06 最后一次提交时间:2022-03-03 19:27:08 得分: 1.00

已提交

设n是描述问题规模的非负整数, 下列程序片段的时间复杂度是 (

A _____ **【正确答案: A】**)

```
x=2;
while(x<n/2)
    x=2*x;
```

- A. $O(\log_2 n)$
- B. $O(n)$
- C. $O(n \log_2 n)$
- D. $O(n^2)$

填空题

1. 首次提交时间:2022-03-03 19:33:54 最后一次提交时间:2022-03-03 19:33:54 得分: 1.00

已提交

下面程序段的运行结果是 600 **【正确答案: 600】**

```
char ch[ ] = "600";
int a, s = 0;
for (a = 0; ch[a] >= '0' && ch[a] <= '9'; a++)
    s = 10 * s + ch[a] - '0';
printf("%d", s);
```

2. 首次提交时间:2022-03-03 19:35:20 最后一次提交时间:2022-03-03 19:35:39 得分: 1.00

已提交

写出下述程序的输出结果 58 **【正确答案: 58】**

```
#include <stdio.h>
int func(int n)
{
    if(n<=1)
        return 1;
    else
        return (2+n*func(n-1));
}

int main()
{
    int x = 4;
    printf("%d\n",func(x));

    return 0;
}
```

3. 首次提交时间:2022-03-03 19:36:08 最后一次提交时间:2022-03-03 19:36:08 得分: 1.00

已提交

给出下述程序的执行结果 8 【**正确答案: 8**】

```
#include <stdio.h>
long fib(int x)
{
    switch(x)
    {
        case 0: return 0;
        case 1:
        case 2: return 1;
    }
    return (fib(x-1)+fib(x-2));
}
int main()
{
    int x=6;
    printf("%d\n",fib(x));

    return 0;
}
```

4. 首次提交时间:2022-03-03 19:37:36 最后一次提交时间:2022-03-03 19:37:46 得分: 1.00

已提交

下面的函数invert的功能是将一个字符串的内容颠倒过来。请将函数补充完整。

k -1 【**正确答案: k -1**】(所有答案填写在此空内, 答案间以空格隔开)

```
void invert (char str [] )
{
    int i,j,填空1____;
    for(i=0,j=strlen(str)填空2_____;i<j;i++,j--)
    {
        k=str[i];
        str[i]=str[j];
        str[j]=k;
    }
}
```

5. 首次提交时间:2022-03-03 19:41:02 最后一次提交时间:2022-03-03 19:41:04 得分: 1.00

已提交

下列程序是否正确, 如果正确给出输出结果, 如果不正确, 答案处填写“不正

确”。不正确 【**正确答案: 不正确**】

```
#include <stdio.h>
void func(char *q)
{
    char a[]="hello";
    q=a;
}
main()
{
    char *p;
    func(p);
    printf("%s\n",p);
}
```

6. 首次提交时间:2022-03-03 19:42:36 最后一次提交时间:2022-03-03 19:43:10 得分: 1.00

已提交

函数index(char s[],char t[])检查字符串s中是否包含字符串t，若包含，则返回t在s中的开始位置(下标值)，否则返回-1。请将函数补充完整。

```
int index(char s[],char t[])
{
    int i,j,k;
    for(i=0;s[i]!='\0';i++)
    {
        for(j=i,k=0;填空1_____ &&s[j]==t[k];j++,k++)
        ;
        if(填空2_____)
            return i;
    }
    return -1;
}
```

_____t[k]!='\0' t[k]=='\0' 【正确答案: t[k]!='\0' t[k]=='\0' 或 s[j]!='\0' t[k]=='\0' 或 t[k]==0 t[k]==0 或 s[j]==0 t[k]==0 或 k<strlen(t) k==strlen(t) 或 s[j]!='\0' &&t[k]!='\0' t[k]=='\0' 或 t[k]!='\0' k==strlen(t) 或 s[j]!='\0' k==strlen(t) 或 s[j]!='\0' &&t[k]!='\0' t[k]=='\0' 或 t[k]!='\0' k==strlen(t) 或 t[k]==0 k==strlen(t) 或 t[k]!='\0' &&s[j]!='\0' t[k]=='\0'】 (所有答案填写在此空内，答案间以空格隔开)

7. 首次提交时间:2022-03-03 19:43:46 最后一次提交时间:2022-03-03 19:44:16 得分: 1.00

已提交

函数squeez(char s[],char c)的功能是删除字符串s中所出现的与变量c相同的字符。

```
void squeez(char s[],char c)
{
    int i,j;
    for(i=j=0;填空1_____;i++)
        if(s[i]==c)
            填空2_____;
        s[j]='\0';
    }
```

_____s[i]!='\0' s[j+1]=s[i] 【正确答案: s[i]!='\0' s[j+1]=s[i] 或 i<strlen(s) s[j+1]=s[i】 (所有答案填写在此空内，答案间以空格隔开)

8. 首次提交时间:2022-03-03 19:46:20 最后一次提交时间:2022-03-03 19:47:32 得分: 1.00

已提交

下面的函数itoah(n,s)完成将无符号十进制整数转换成十六进制表示，并存入字符串数组s中。程序中用到的函数reverse(char s[])是一个将字符串置逆的函数。

```
void reverse(char s[])
{
    int temp,i,j;
    for(i=0,j=strlen(s)-1;i<j;i++,j--)
    {
        temp=s[i];
        s[i]=s[j];
        s[j]=temp;
    }
}

void itoh(unsigned n,char s[])
{
    int h,i=0;
    do{
        h=n%16;
        s[i++]=(h<=9)?h+'0':h+'A'_____;
    }while((n/=16)!=0);
    _____;
    reverse(s);
}
```

_____ -10 s[i]='\0' 【正确答案: -10 s[i]='\0' 或 -10 s[i]=0】 (所有答案填写在此空内，答案间以空格隔开)

9. 首次提交时间:2022-03-03 19:57:26 最后一次提交时间:2022-03-08 23:17:52 得分: 1.00

已提交

下面程序的功能是将字符串 s 中的内容按逆序输出,但不改变串中的内容,
请选择填空(答案请填写所选择的两个字母,中间用空格格开,如A A的形式) : **C B** 【 正确答案: c b】

```
#include <stdio.h>
void inverp(char *a)
{
    if ( _____ )
        return;
    inverp(a+1);
    printf("%c", _____ );
}
int main()
{
    char s[10] = "hello!";
    inverp(s);

    return 0;
}
```

Selection 1 :

- (A) *a != '\0'
- (B) *a != NULL
- (C) *a == '\0'
- (D) !a * == 0

Selection 2 :

- (A) * (a-1)
- (B) *a
- (C) * (a+1)
- (D) * (-a)

10. 首次提交时间:2022-03-03 19:58:29 最后一次提交时间:2022-03-03 20:00:16 得分: 1.00

已提交

下面程序的功能是将已按升序排好的两个字符串a和b中的字符按升序并归到字符串c中。请选择填空将程序补充完整: A D A 【**正确答案: A D A**】(所有答案填写在此空内, 答案间以空格隔开)

```
#include<stdio.h>
#include<string.h>
int main()
{
    char a[]="acegikm";
    char b[]="bdfhjlnpq";
    char c[80],*p;
    int i=0,j=0,k=0;
    while(a[i]!='\0'&&b[j]!='\0')
    {
        if(a[i]<b[j]) {_____}
        else{_____}
    }
    c[k]='\0';
    if(_____)p=b+j;
    else p=a+i;
    strcat(c,p);
    puts(c);

    return 0;
}
```

填空1:

- (A)c[k++]=a[i++];
- (B)c[k++]=a[j++];
- (C)c[k++]=b[i++];
- (D)c[k++]=b[j++];

填空2:

- (A)c[k++]=a[i++];
- (B)c[k++]=a[j++];
- (C)c[k++]=b[i++];
- (D)c[k++]=b[j++];

填空3:

- (A)a[i]=='\0'
- (B)a[j]=='\0'
- (C)b[i]=='\0'
- (D)b[j]=='\0'

编程题

#	题目	分值	批阅信息
---	----	----	------

#	题目	分值	批阅信息												
1.	<p>扩展字符A</p> <p>【问题描述】 从键盘输入包含扩展符'-'的字符串，将其扩展为等价的完整字符，例如将a-d扩展为abcd，并输出扩展后的字符串。</p> <p>要求：只处理[a-z]、[A-Z]、[0-9]范围内的字符扩展，即只有当扩展符前后的字符同时是小写字母、大写字母或数字，并且扩展符后的字符大于扩展符前的字符时才进行扩展，其它情况不进行扩展，原样输出。例如：a-R、D-e、0-b、4-B等字符串都不进行扩展。</p> <p>【输入形式】 从键盘输入包含扩展符的字符串</p> <p>【输出形式】 输出扩展后的字符串</p> <p>【输入样例1】 ADEa-g-m02</p> <p>【输出样例1】 ADEabcdefgijklm02</p> <p>【输入样例2】 cdeT-bcd</p> <p>【输出样例2】 cdeT-bcd</p> <p>【样例说明】 将样例1的输入ADEa-g-m02扩展为：ADEabcdefgijklm02；样例2的输入cdeT-bcd中，扩展符前的字符为大写字母，扩展符后的字符为小写字母，不在同一范围内，所以不进行扩展。</p> <p>【评分标准】 该题要求扩展字符，提交程序文件expand.c。</p>	20.00	<p>下载源文件</p> <p>得分20.00 最后一次提交时间:2022-03-03 20:10:14 成功编译,但有警告信息. expand.c: In function 'main': expand.c:20:45: warning: suggest parentheses around '&&' within ' ' [-Wparentheses] if(s[i-1]>='a'&&s[i-1]<='z'&&s[i+1]>='a'&&s[i+1]<='z' ^ expand.c:22:48: warning: suggest parentheses around '&&' within ' ' [-Wparentheses] s[i-1]>='0'&&s[i-1]<='9'&&s[i+1]>='0'&&s[i+1]<='9' ^</p> <p>共有测试数据:5 平均占用内存:1.397K 平均CPU时间:0.00694S 平均墙钟时间:0.00692S</p> <table border="1"> <thead> <tr> <th>测试数据</th> <th>评判结果</th> </tr> </thead> <tbody> <tr> <td>测试数据1</td> <td>完全正确</td> </tr> <tr> <td>测试数据2</td> <td>完全正确</td> </tr> <tr> <td>测试数据3</td> <td>完全正确</td> </tr> <tr> <td>测试数据4</td> <td>完全正确</td> </tr> <tr> <td>测试数据5</td> <td>完全正确</td> </tr> </tbody> </table>	测试数据	评判结果	测试数据1	完全正确	测试数据2	完全正确	测试数据3	完全正确	测试数据4	完全正确	测试数据5	完全正确
测试数据	评判结果														
测试数据1	完全正确														
测试数据2	完全正确														
测试数据3	完全正确														
测试数据4	完全正确														
测试数据5	完全正确														

[详细](#)

#	题目	分值	批阅信息
2.	表达式计算（支持空格，连乘，连除）	20.00	下载源文件
【问题描述】			
从标准输入中读入一个整数算术运算表达式，如 $5 - 1 * 2 * 3 + 12 / 2 / 2 =$ 。计算表达式结果，并输出。			
要求：			
1、表达式运算符只有+、-、*、/，表达式末尾的 '=' 字符表示表达式输入结束，表达式中可能会出现空格；			
2、表达式中不含圆括号，不会出现错误的表达式；			
3、出现除号/时，以整数相除进行运算，结果仍为整数，例如： $5/3$ 结果应为1。			
【输入形式】			
在控制台中输入一个以'='结尾的整数算术运算表达式。			
【输出形式】			
向控制台输出计算结果（为整数）。			
【样例1输入】			
$5 - 1 * 2 * 3 + 12 / 2 / 2 =$			
【样例1输出】			
2			
【样例2输入】			
500 =			
【样例2输出】			
500			
【样例1说明】			
输入的表达式为 $5 - 1 * 2 * 3 + 12 / 2 / 2 =$ ，按照整数运算规则，计算结果为2，故输出2。			
【样例2说明】			
输入的表达式为 $500 =$ ，没有运算符参与运算，故直接输出500。			
算法之一提示：			
1、可以利用gets函数，读取整行表达式；			
2、对于空格，可以考虑首先去除表达式中的所有空格			
3、可以设一计数器用来记录已读取、但未参加运算的运算符的个数，根据该计数器来判断如何进行运算；			
4、可以设计一函数：实现二元整数算术运算。			
【评分标准】			
该题要求输出整数表达式的计算结果，共有5个测试点。上传C语言文件名为example1c.c。			
详细			

#	题目	分值	批阅信息
3.	小数形式与科学计数法转换（简）	20.00	下载源文件
【问题描述】			
编写一个程序，将用小数表示的浮点数，转换成科学计数法的形式输出。输入的数据没有符号，小数点前后必有数字，且全为有效数据，即小数点后的末尾数字不为0；小数点前若只有一位数字，可以为0，否则小数点前的最高位数字不为0。			
提示：以字符串形式保存相关数据。			
【输入形式】			
从控制台输入一小数，最后有回车换行符，所有输入的字符数不会超过100。			
【输出形式】			
以科学计数法形式输出数据。输出的数据由以下几部分构成： 1.底数部分是一个小数或整数， 若为小数，则小数点前后必有数字，而且都为有效数字。 即：小数点前只有一位大于0的数字，小数点后的末尾数字不能为0。若为整数，则只有一位数字，不带小数点。 2.必有小写字母“e”。 3.指数部分是一个整数，若大于等于0，则不带正号“+”。若小于0，则需要带负号“-”，且整数的最高位数字不为0。			
【输入样例1】			
0.0000000000000002			
【输出样例1】			
2e-15			
【输入样例2】			
8.9845623489651700659			
【输出样例2】			
8.9845623489651700659e0			
【输入样例3】			
3672985999990934534903948595095686597 95603.4			
【输出样例3】			
3.672985999990934534903948595095686597 956034e41			
【样例说明】			
以小数形式输入数据，然后转换成科学计数法形式输出。			
【评分标准】			
该题要求以科学计数法形式输出数据，提交程序文件名为notation.c。			

#	题目	分值	批阅信息												
4.	<p>超长正整数的减法</p> <p>【问题描述】 编写程序实现两个超长正整数（每个最长80位数字）的减法运算。</p> <p>【输入形式】 从键盘读入两个整数，要考虑输入高位可能为0的情况（如00083）。</p> <p>1. 第一行是超长正整数A； 2. 第二行是超长正整数B；</p> <p>【输出形式】 输出只有一行，是长整数A减去长整数B的运算结果，从高到低依次输出各位数字。要求：若结果为0，则只输出一个0；否则输出的结果的最高位不能为0，并且各位数字紧密输出。</p> <p>【输入样例】</p>	20.00	<p>下载源文件</p> <p>得分20.00 最后一次提交时间:2022-03-03 21:05:56 共有测试数据:5 平均占用内存:1.396K 平均CPU时间:0.00652S 平均墙钟时间:0.00648S</p> <table border="1"> <thead> <tr> <th>测试数据</th> <th>评判结果</th> </tr> </thead> <tbody> <tr> <td>测试数据1</td> <td>完全正确</td> </tr> <tr> <td>测试数据2</td> <td>完全正确</td> </tr> <tr> <td>测试数据3</td> <td>完全正确</td> </tr> <tr> <td>测试数据4</td> <td>完全正确</td> </tr> <tr> <td>测试数据5</td> <td>完全正确</td> </tr> </tbody> </table>	测试数据	评判结果	测试数据1	完全正确	测试数据2	完全正确	测试数据3	完全正确	测试数据4	完全正确	测试数据5	完全正确
测试数据	评判结果														
测试数据1	完全正确														
测试数据2	完全正确														
测试数据3	完全正确														
测试数据4	完全正确														
测试数据5	完全正确														

[详细](#)

234098
134098703578230056

【输出样例】
- 134098703577995958

【样例说明】
进行两个正整数减法运算， 234098 - 134098
703578230056 = - 134098703577995958。

【评分标准】
完全正确得20分，每个测试点4分，提交程序
文件名为subtract.c。

#	题目	分值	批阅信息
5.	全排列数的生成	20.00	下载源文件
	<p>【问题描述】输入整数N(1 <= N <= 10), 生成从1~N所有整数的全排列。</p> <p>【输入形式】输入整数N。</p> <p>【输出形式】输出有N!行, 每行都是从1~N所有整数的一个全排列, 各整数之间以空格分隔。各行上的全排列不重复。输出各行遵循“小数优先”原则, 在各全排列中, 较小的数尽量靠前输出。如果将每行上的输出看成一个数字, 则所有输出构成升序数列。具体格式见输出样例。</p> <p>【样例输入1】1</p> <p>【样例输出1】1</p> <p>【样例说明1】输入整数N=1, 其全排列只有一种。</p> <p>【样例输入2】3</p> <p>【样例输出2】</p> <pre>1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1</pre> <p>【样例说明2】输入整数N=3, 要求整数1、2、3的所有全排列, 共有N!=6行。且先输出1开头的所有排列数, 再输出2开头的所有排列数, 最后输出3开头的所有排列数。在以1开头的所有全排列中同样遵循此原则。</p> <p>【样例输入3】10</p> <p>【样例输出3】</p> <pre>1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 10 9 1 2 3 4 5 6 7 9 8 10 1 2 3 4 5 6 7 9 10 8 1 2 3 4 5 6 7 10 8 9 1 2 3 4 5 6 7 10 9 8 1 2 3 4 5 6 8 7 9 10 1 2 3 4 5 6 8 7 10 9 1 2 3 4 5 6 8 9 7 10 1 2 3 4 5 6 8 9 10 7 【样例说明3】输入整数N=10, 要求整数1、2、3、.....10的所有全排列。上例显示了输出的前10行。</pre> <p>【运行时限】要求每次运行时间限制在20秒之内。超出该时间则认为程序错误。提示: 当N增大时, 运行时间将急剧增加。在编程时要注意尽量优化算法, 提高运行效率。</p> <p>【评分标准】该题要求输出若干行整数。。</p>	得分20.00 最后一次提交时间:2022-03-03 21:09:18 共有测试数据:5 平均占用内存:1.396K 平均CPU时间:0.00526S 平均墙钟时间:0.00523S	测试数据 评判结果
		测试数据1	完全正确
		测试数据2	完全正确
		测试数据3	完全正确
		测试数据4	完全正确
		测试数据5	完全正确

[详细](#)

北京航空航天大学

若重置密码, 请与当前的任课教师联系