**张一大西农复试机考分享**

**水平有限，难免有不足之处，还请与我交流。**

**联系方式：Email：zhang\_yi\_da@foxmail.com**

1.按yyyy-mm-dd格式输入一个**日期**，判断这是这一年的第几天

2.筛选法求(1~n)**素数**

分析:由希腊著名数学家埃拉托色尼提出的所谓“筛法”，步骤如下：

①将所有候选数放入筛中；

②找筛中最小数（必为素数）next，放入集合primes中；

③将next的所有倍数从筛中筛去；

④重复②～④直到筛空。编程时，用集合变量sieve表示筛子，用集合primes存放所有素数。

3. 连续输入三个**字符串**，在第一个字符串中查找第二个字符串，如果存在，就将之替换为第三个字符串。

4.C语言编程序，打印前10对**孪生素数**。如果两个素数之差为2，则称他们为孪生素数。如3和5是孪生素数，5和7是孪生素数.

5.产生10个1~21的随机数，要求**随机数**各不相同。随后对它排序，输出**排序**后的结果

6.一个文件**读取数据**，进行排序后去掉相同的数据后，放在另一个文件；

in.dat存放的数据类型，第一行是数据的总数，第二行是存放的是数据本身。如：

10   
0  2  5  7  2  5  4  8  3  12

将上面的数据进行排序，并去掉上面相同的数据，并放在out.dat文件中，第一行是数据的总和，第二行是数据本身。

7.从**文本文件**中导入班级学生信息：学号、姓名、性别、籍贯，显示导入的学生信息

将学号重复的删除，按学号、姓名、性别、籍贯相等和不相等查找。数据见附录class.txt

8.在1个**字符串**中找到第1个只出现1次的字符。

可考虑建立想对于字符数组的计数数组，长度相等。从第一个字符开始和后面字符依次比对，若有相同则两相对应的计数数组全为1，继续。最后判别计数

9.在一个字**符数组**中查找一个指定的字符，若数组中含有该字符，则输出该字 符在数组第一次出现的位置(可能为假题，如此简单！！！)

10.**文件压缩**

描述：通过文件压缩程序，可将一个文件压缩为一个新文件，压缩文件比原文件要小。设原文件是由‘0’和‘1’两种字符组成的二维数组（见输入样例），其中第1行给定了数组的行数和列数，压缩算法按重复出现字符次数代替原文件中连续重复的字符。例如：”0000011110000000111100000”，由于依次出现5个0、4个1、7个0、4个1和5个0，因此可用”5 4 7 4 5”表示。要求实现将输入样例所示文本文件(A.txt)压缩为一个新文件(B.txt)。数据见附录。

11.算身份证

身份证的最后一位是根据前17位数字计算出来的**检验码**。计算方法是：将身份证号码前17位数分别乘以不同的系数。从第1位到第17位的系数分别为：7 9 10 5 8 4 2 1 6 3 7 9 10 5 8 4 2；将乘积之和除以11，余数可能为0 1 2 3 4 5 6 7 8 9 10。则根据余数，分别对应的最后一位身份证的号码为1 0 X 9 8 7 6 5 4 3 2。编写程序，输入身份证号码前17位，输出对应的检验码。

12.打印**月历**

首先需要确定该年是闰年还是平年，闰年的二月是29天，全年是366天；平年二月是28天，全年365天。还需要确定该年的每一天是星期几。从1900年1月1日是星期一开始，该值可以通过使用该年到1900年的总天数模7来求得。最后考虑怎样输出日历，可以先将日历存储在一个二维数组里面，也可以直接将它输出。

13.1~9组成三个3位的**平方数**       
    将1、2、3、4、5、6、7、8、9九个数字分成三组，每个数字只能用一次，即每组三个数不允许有重复数字，也不许同其它组的三个数字重复，要求每组中的三位数都组成一个平方数。

14.描述：创建一个带头结点的**单链表**，在单链表中删除值相同的多余结点，并遍历链表，删除链表最大节点。

输入：从键盘上输入数据元素个数n和对应n个元素。

输出：输出删除多余结点的单链表序列。

输入样例：6 3 1 3 4 4 5

输出样例：3 1 4 5

15.**质因数**分解：把一个合数分解成若干个质因数的乘积的形式，即求质因数的过程叫做分解质因数。

16.一元四次方程，求**近似解**。求解一元四次方程x^4 - 625\*x^3 - 2\*x^2 + 1 = 0求 0 到 1 之间的那个解,

17.两数组合并到新数组，然后新数组元素排序输出。

1. 张一大代码：

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

typedef int bool;

#define TRUE 1

#define FALSE 0

#define SIZE 12

bool leap(int year);

int main()

{

//按yyyy-mm-dd格式输入一个日期，判断这是这一年的第几天

int year,month,day;

int a[SIZE] = {31,28,31,30,31,30,31,31,30,31,30,31};

int sum = 0;

int i;

printf("请输入年月日：xxxx年xx月xx日");

scanf("%d年%d月%d日",&year,&month,&day);

if(leap(year)==TRUE){

a[1] = 29;

}

for(i=0;i<month-1;i++){

sum = sum+a[i];

}

sum = sum+day;

printf("%-d",sum);

return 0;

}

bool leap(int year){

if(year%400==0 || (year%4==0&&year%100!=0)){

return TRUE;

}

return FALSE;

}

2.张一大代码

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#define SIZE 100

void prime\_select(int a[]);

int main()

{

//筛选法求(1~n)素数

//分析:由希腊著名数学家埃拉托色尼提出的所谓“筛法”，步骤如下：

//①将所有候选数放入筛中；

//②找筛中最小数（必为素数）next，放入集合primes中；

//③将next的所有倍数从筛中筛去；

//④重复②～④直到筛空。编程时，用集合变量sieve表示筛子，

//用集合primes存放所有素数。

int a[SIZE];

int i,j;

int sieve;

//初始化

for(i=0;i<SIZE;i++){

a[i] = i+1;

}

//找到最小筛子

a[0] = 0;

//使得筛子的倍数置0

//重新选择最小数为筛子

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

if(a[i]>0){

sieve = a[i];

for(j=i+1;j<SIZE;j++)

{

if(a[j]%sieve==0)

{

a[j] = 0;

}

}

}

}

//输出

for(i=0;i<SIZE;i++){

if(a[i]>0){

printf("%-3d",a[i]);

}

}

return 0;

}

3.张一大代码：

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#define SIZE 30

void prime\_select(int a[]);

int main()

{

//连续输入三个字符串，在第一个字符串中查找第二个字符串，

//如果存在，就将之替换为第三个字符串。

char chara1[SIZE] = {"abcdefgh"},chara2[SIZE]={"def"},chara3[SIZE]={"he"};

char \*ptr;

int i,j=0,k;

int len;

int flag = 0;

//返回字符串2在字符串1中的首地址

ptr = strstr(chara1,chara2);

printf("%s,%d\n",ptr,strlen(ptr));

//比较子字符串和替换字符串的长度

len = strlen(chara2) - strlen(chara3);

if(len>0){

//先缩短

for(i=0;i<strlen(ptr);i++){

ptr[i] = ptr[i+len];

}

for(i=0; i<strlen(chara3); i++)

{

ptr[i] = chara3[i];

}

}else{

len = -len; //使得长度为正数

//延长有缺陷，看谁可以啊

//现用挨个比对

for(i=0;i<strlen(chara1);i++){

flag = 0; //标记是否符合情况

if(chara1[i]==chara2[j]){

for(k=1;k<strlen(chara2);k++){

if(chara1[i+1]!=chara2[j+1]){

flag = 1;

}

else{

i++;

j++;

}

}

//对以上循环的判断

if(flag==1){

i--;

continue;

}

else{

//找到相同字符串，后移。

for(i=strlen(chara1)+len-1;i>len;i--){

chara1[i] = chara1[i-len];

}

chara1[strlen(chara1)+len] = '\0';

//字符3复制到字符1

for(i=0; i<strlen(chara3); i++)

{

ptr[i] = chara3[i];

}

break;

}

}

}

}

printf("%s",chara1);

return 0;

}

4.张一大代码：

#include <stdio.h>

#include <stdlib.h>

#include<time.h>

#define SIZE 150

void prime\_select(int a[]);

int main()

{

//C语言编程序，打印前10对孪生素数。如果两个素数之差为2，

//则称他们为孪生素数。如3和5是孪生素数，5和7是孪生素数.

int a[SIZE];

int i;

//初始化

for(i=0;i<SIZE;i++){

a[i] = i+1;

}

prime\_select(a);

return 0;

}

//筛选素数

void prime\_select(int a[]){

int i,j;

int temp;

a[0] = 0;

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

if(a[i]!=0){

temp = a[i];

}

else{

continue;

}

//把temp的倍数置0

for(j=i+1;j<SIZE;j++){

if(a[j]%temp == 0){

a[j] = 0;

}

}

}

j = 1;

for(i=2;i<SIZE;i++){

if(a[i]>0){

//判断前两者之差

if(a[i]-a[j]==2){

printf("%-d,%-d\n",a[j],a[i]);

}

else{

j = i;

}

}

}

}

5.张一大代码：

#include <stdio.h>

#include <stdlib.h>

#include<time.h>

#define SIZE 10

int main()

{

// 产生10个1~21的随机数，要求随机数各不相同。随后对它排序，输出排序后的结果

int i,j;

int temp;

int a[SIZE];

srand(time(NULL));

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

a[i] = rand()%21+1;

//判断是否相等

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

if(a[i] == a[j]){

i--;

break;

}

}

}

for(i=0;i<SIZE;i++){

printf("%-3d",a[i]);

}

//冒泡法排序

for(i=0;i<SIZE-1;i++){

for(j=0;j<SIZE;j++){

if(a[j]>a[j+1]){

temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

}

}

}

printf("\n");

for(i=0;i<SIZE;i++){

printf("%-3d",a[i]);

}

return 0;

}

6.张一大代码：

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#define SIZE 10

int main()

{

//一个文件读取数据，进行排序后去掉相同的数据后，放在另一个文件；

//in.dat存放的数据类型，第一行是数据的总数，第二行是存放的是数据本身。如：

//10

//0 2 5 7 2 5 4 8 3 12

//将上面的数据进行排序，并去掉上面相同的数据，并放在out.dat文件中，第一行是数据的总和，第二行是数据本身。

FILE \*fp;

int num;

int temp;

int i,j,k;

int a[SIZE];

fp = fopen("in.dat","r");

//打开文件要判断

if(fp == NULL){

printf("The file can't open!");

exit(EXIT\_SUCCESS);

}

//读取数据总数

fscanf(fp,"%d\n",&num);

for(i=0;i<num;i++){

fscanf(fp,"%d",&a[i]);

printf("%-3d",a[i]);

}

printf("\n");

//删除重复的

for(i=0;i<num;i++){

temp = a[i];

printf("%d,",temp);

for(j=i+1;j<num;j++){

if(temp == a[j]){

//删除

for(k=j;k<num-1;k++){

a[k] = a[k+1];

}

num--;

}

}

}

//冒泡法排序

for(i=0;i<num-1;i++){

for(j=0;j<num;j++){

if(a[j]>a[j+1]){

temp = a[j];

a[j] = a[j+1];

a[j+1] = temp;

}

}

}

printf("\n");

for(i=0;i<num;i++){

printf("%-3d",a[i]);

}

fclose(fp);

return 0;

}

7.张一大代码：

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#define SIZE 10

#define LENGTH 61

struct Student{

char number[SIZE];

char name[SIZE];

char sex[SIZE];

char location[SIZE];

}stud[LENGTH];

int main()

{

//从文本文件中导入班级学生信息：学号、姓名、性别、籍贯，显示导入的学生信息

//将学号重复的删除，按学号、姓名、性别、籍贯相等和不相等查找。数据见附录class.txt

FILE \*fp;

int i,j,k;

char character[SIZE];

int cnt=0;

fp = fopen("class.txt","r");

//打开文件判断是否成功

if(fp==NULL){

printf("The file can not open!");

exit(EXIT\_SUCCESS);

}

//读入数据，显示信息

for(i=0;i<LENGTH;i++){

fscanf(fp,"%s%s%s%s\n",stud[i].number,stud[i].name,stud[i].sex,stud[i].location);

printf("%s,%s,%s,%s\n",stud[i].number,stud[i].name,stud[i].sex,stud[i].location);

}

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

//删除重复的

for(i=0;i<LENGTH;i++){

strcpy(character,stud[i].number);

for(j=i+1;j<LENGTH;j++){

if(strcmp(character,stud[j].number)==0){

//重复的输出一下

printf("%s,%s,%s,%s\n",stud[j].number,stud[j].name,stud[j].sex,stud[j].location);

cnt++; //统计重复的个数

for(k=j;k<LENGTH;k++){

strcpy(stud[k].location,stud[k+1].location);

strcpy(stud[k].name,stud[k+1].name);

strcpy(stud[k].number,stud[k].number);

strcpy(stud[k].sex,stud[k+1].sex);

}

}

}

}

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

for(i=0;i<LENGTH-cnt;i++){

// fscanf(fp,"%s%s%s%s\n",stud[i].number,stud[i].name,stud[i].sex,stud[i].location);

printf("%s,%s,%s,%s\n",stud[i].number,stud[i].name,stud[i].sex,stud[i].location);

}

fclose(fp);

return 0;

}

8.张一大代码：

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#define SIZE 30

int main()

{

//在1个字符串中找到第1个只出现1次的字符

char character[SIZE] = {"aabccbdc"};

char ch;

int i,j;

int cnt;

int count = strlen(character); //字符串长度

// printf("%d\n",count);

//从第一个字符往后比对，找到相同的再从下一个字符比对，否则找到

for(i=0;i<count;i++){

ch = character[i];

// printf("%c",ch);

cnt = 0;

for(j=0;j<count;j++){

if(character[j] == ch){

cnt++;

if(cnt==2){ //cnt为2表示有相同的元素

break;

}

}

}

if(cnt==1){

printf("%c",ch);

break;

}

else{

continue;

}

}

return 0;

}

9.张一大代码：

#include <stdio.h>

#include <stdlib.h>

#include<string.h>

#define SIZE 30

int main()

{

//在一个字符数组中查找一个指定的字符，若数组中含有该字符，

//则输出该字符在数组第一次出现的位置

char characer[SIZE] = {"abcdefghijklmnopqrestuvwxya"};

char ch = 'c';

int i;

int flag = 0;

for(i=0;i<strlen(characer);i++){

if(ch == characer[i]){

printf("%d",i+1);

flag = 1;

}

}

if(flag == 0){

printf("%c is not exist in this string!",ch);

}

return 0;

}

10.张一大代码：

#include <stdio.h>

#include <stdlib.h>

#define SIZE 30

int main()

{

/\*//文件压缩

//描述：通过文件压缩程序，可将一个文件压缩为一个新文件，压缩文件比原文件要小。

//设原文件是由‘0’和‘1’两种字符组成的二维数组（见输入样例），其中第1行给定了数组的行数和列数，

//压缩算法按重复出现字符次数代替原文件中连续重复的字符。例如：”0000011110000000111100000”，

//由于依次出现5个0、4个1、7个0、4个1和5个0，因此可用”5 4 7 4 5”表示。

//要求实现将输入样例所示文本文件(A.txt)压缩为一个新文件(B.txt)。

\*/

//先打开文件，获取行数和列数

FILE \*fp,\*fp2;

int row,col;

int i,j;

char c[SIZE][SIZE]; //定义字符串数组，不敢定义成字符串

char ch;

int cnt; //计数

fp = fopen("A.txt","r");

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

//打开文件判断是否成功

if(fp == NULL || fp2 == NULL){

printf("The file can't open!");

exit(EXIT\_SUCCESS);

}

//提取行数和列数

fscanf(fp,"%d%d\n",&row,&col);

printf("%d,%d\n",row,col);

//用字符串数组提取所有0和1

for(i=0;i<row;i++){

fscanf(fp,"%s",c[i]);

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

}

//对提取出的字符进行压缩，输出到B.txt中

for(i=0;i<row;i++){

ch = c[i][0]; //每行第一字符赋给ch

cnt = 1; //ch算有一个字符

for(j=1;j<col;j++){ //从第二个字符开始比较

if(ch == c[i][j]){

cnt++;

}

else{

fprintf(fp2,"%-3d",cnt);

printf("%-3d",cnt);

cnt = 1;

ch = c[i][j];

}

//到达最后一个字符，输出

if(j == col-1)

{

fprintf(fp2,"%-3d",cnt);

printf("%-3d",cnt);

}

}

fputc('\n',fp2);

printf("\n");

}

fclose(fp);

fclose(fp2);

return 0;

}

11.张一大的代码：

#include <stdio.h>

#include <stdlib.h>

bool leap(int year);

int main()

{

//11.算身份证

//身份证的最后一位是根据前17位数字计算出来的检验码。计算方法是：

//将身份证号码前17位数分别乘以不同的系数。从第1位到第17位的系数分别为：

//7 9 10 5 8 4 2 1 6 3 7 9 10 5 8 4 2；将乘积之和除以11，

//余数可能为0 1 2 3 4 5 6 7 8 9 10。则根据余数，

//分别对应的最后一位身份证的号码为1 0 X 9 8 7 6 5 4 3 2。

//编写程序，输入身份证号码前17位，输出对应的检验码。

int weight[17] = {7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2};

int num[18];

int sum = 0;

int i;

int last[11] = {1,0,'x',9,8,7,6,5,4,3,2};

printf("请输入你身份证号码的前17位(空格隔开)：");

for(i=0;i<17;i++){

scanf("%d",&num[i]);

}

for(i=0;i<17;i++){

sum = sum + num[i]\*weight[i];

}

//余数作为下标进行计算

num[17] = last[sum%11];

//输出

if(sum%11!=2){

for(i=0; i<18; i++)

{

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

}

}

else{

printf("%c",last[2]);

}

return 0;

}

12.张一大的代码：

#include <stdio.h>

#include <stdlib.h>

#define TRUE 1

#define FALSE 0

typedef int bool;

bool leap(int year);

int main()

{

/\*//12.打印月历

//首先需要确定该年是闰年还是平年，闰年的二月是29天，全年是366天；

//平年二月是28天，全年365天。还需要确定该年的每一天是星期几。

//从1900年1月1日是星期一开始，该值可以通过使用该年到1900年的总天数模7来求得。

//最后考虑怎样输出日历，可以先将日历存储在一个二维数组里面，也可以直接将它输出。\*/

int year = 2018,month = 3;

int a[12] = {31,28,31,30,31,30,31,31,30,31,30,31};//a数组存储每月天数

int sum\_day; //总天数

int week;

int i,j=1; //j从1~月份的终止天数

if(leap(year)==TRUE){

a[1] = 29;

}

sum\_day = (year - 1900)\*365 + (year-1-1900)/4;

for(i=0;i<month-1;i++){

sum\_day = sum\_day + a[i];

}

sum\_day = sum\_day + 1; //每个月的第一天

week = sum\_day%7; //周几

printf("日 一 二 三 四 五 六\n");

for(i=0;i<week;i++){

printf(" ");

}

for(i=week;i<7;i++){

printf("%-3d",j++);

}

printf("\n");

for(i=1;i<=a[month-1]-(7-week);i++){

printf("%-3d",j++);

if(i%7==0)

printf("\n");

}

return 0;

}

//判断是否闰年

bool leap(int year){

if(year%400==0 || (year%4==0&&year%100==0)){

return TRUE;

}

else{

return FALSE;

}

}

13.张一大代码：

#include <stdio.h>

#include <stdlib.h>

int main()

{/\*

//1~9组成三个3位的平方数

// 将1、2、3、4、5、6、7、8、9九个数字分成三组，每个数字只能用一次，

//即每组三个数不允许有重复数字，也不许同其它组的三个数字重复，要求每组中的

//三位数都组成一个平方数。

//分析最小的平方数为123，最大的987,平方根的范围在12~31，平方数中不含0；\*/

int i;

int a[9] = {0};

int unit,decade,hundred;

int cnt = 1; //标记是否输出了三次

int begin = 12+2+1+1+1+1+1;

while(cnt!=4){

for(i=begin; i<=32; i++)

{

hundred = i\*i/100; //百位

decade = i\*i/10%10; //十位

unit = i\*i%10; //各位

if(hundred==0 || decade==0 || unit==0)

{

continue;

}

if(hundred==decade || decade==unit || hundred==unit)

{

continue;

}

//第二次

if(cnt==2)

{

if(a[hundred-1]==1||a[decade-1]==1||a[unit-1]==1)

{

continue;

}

printf("第二次\n");

}

//第三次

if(cnt==3)

{

if((a[hundred-1]==0)&&(a[decade-1]==0)&&(a[unit-1]==0))

{

printf("第三次\n");

printf("%-3d",i\*i);

cnt++;

// printf("%d",cnt);

break;

}

else{

continue;

}

}

//第一次

a[hundred-1] = 1; //a数组标记是否使用过

a[decade-1] = 1;

a[unit-1] = 1;

cnt++;

printf("%-3d\n",i\*i);

}

}

return 0;

}

14.张一大代码：

#include <stdio.h>

#include <stdlib.h>

//单链表结构体

typedef struct LNode{

int data;

struct LNode \*next;

}LNode,\*LinkList;

void create\_list(LNode \*L);

void delete\_same(LNode \*L);

void delete\_max(LNode \*L);

int main()

{

//创建一个带头结点的单链表，在单链表中删除值相同的多余结点，并遍历链表，删

//除链表最大节点。

//输入：从键盘上输入数据元素个数n和对应n个元素。

//输出：输出删除多余结点的单链表序列。

//输入样例：6 3 1 3 4 4 5

//输出样例：3 1 4 5

LinkList L = (LinkList)malloc(sizeof(LNode));

L->next = NULL;

create\_list(L);

printf("\n");

delete\_same(L);

printf("\n");

delete\_max(L);

return 0;

}

//创建带头结点的单链表

void create\_list(LNode \*L){

int n;

int i;

LNode \*s,\*p = L;

//尾插法实现初始单链表

scanf("%d",&n);

for(i=0;i<n;i++){

s = (LinkList)malloc(sizeof(LNode));

scanf("%d",&(s->data));

// printf("%-3d",s->data);

s->next = NULL;

p->next = s;

p = s;

}

//输出(相当于遍历链表了)

p = L->next;

while(p!=NULL){

printf("%-3d",p->data);

p = p->next;

}

}

//在链表里删除相同的元素

void delete\_same(LNode \*L){

LNode \*p = L->next,\*s,\*r;

while(p!=NULL){

r = p; //存储s的前一元素

s = p->next; //从p的后一元素开始循环

while(s!=NULL){

if(s->data == p->data){

r->next = s->next;

free(s);

s = r->next;

}

else{

r = s;

s = s->next;

}

}

p = p->next;

}

//输出删除后的元素

p = L->next;

while(p!=NULL){

printf("%-3d",p->data);

p = p->next;

}

}

//删除最大结点

void delete\_max(LNode \*L){

LNode \*p = L->next,\*s,\*r;

int temp;

//先用选择排序，删除最后的元素

while(p!=NULL){

r = p;

s = p->next;

while(s!=NULL){

if(r->data>s->data){

r = s;

}

s = s->next;

}

//这里交换数据，结点不动

if(r != p){

temp = p->data;

p->data = r->data;

r->data = temp;

}

p = p->next;

}

//找到最大结点，删除释放，前一结点next域置空

p = L->next;

s = p->next;

while(s!=NULL){

r = p;

p = p->next;

s = p->next;

}

free(p);

r->next = NULL;

//输出

p = L->next;

while(p!=NULL){

printf("%-3d",p->data);

p = p->next;

}

}

15. 张一大代码：

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

#define SIZE 100

void prime(int n,int a[]);

void prime\_factor(int n);

int j=0;

int primes[SIZE];

int main()

{

//把一个合数分解成若干个质因数的乘积的形式，即求质因数的过程叫做分解质因数。

int n = SIZE;

int a[SIZE];

prime(n,a);

prime\_factor(36);

return 0;

}

//求合数的素因子

void prime\_factor(int n){

int i;

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

//先循环除以第一个素数

while(n%primes[i]==0){

printf("%-3d",primes[i]);

n = n/primes[i];

}

}

}

//判断素数

void prime(int n,int a[]){

int i;

int seive; //定义筛子

//a数组存放1~SIZE

a[0] = 0;

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

a[i] = i+1;

}

//首先筛子是2

seive = 2;

while(1){

//找到筛子的倍数置0

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

{

if(a[i]%seive==0)

{

a[i] = 0;

}

}

//将素数全部放到primes数组中

primes[j] = seive; //j是素数数组的下标

j++;

//把不为0的最小数赋给筛子

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

{

if(a[i]>seive)

{

seive = a[i]; //

break;

}

}

if(i>=n-1) break;//跳出循环的条件

}

//输出1-SIZE的素数

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

{

printf("%-3d",primes[i]);

}

printf("\n");

}

答案：

16.张一大代码：

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

double func(double x);

int main()

{

//求解一元四次方程x^4 - 625\*x^3 - 2\*x^2 + 1 = 0求 0 到 1 之间的那个解,

//方法是二分法逼近思想

double low = 0,high = 1;

double mid = 0;

double absolute;

absolute = fabs(func(high)-func(low)); //比较两个函数值的距离，距离小于1e-5时输出结果

while(absolute>1e-5){

mid = (low + high)/2; //取中值

//确保乘积小于0

if(func(low)\*func(mid)<0){

high = mid;

}

else if(func(mid)\*func(high)<0){

low = mid;

}

absolute = fabs(func(high)-func(low));

}

printf("The result is %lf,方程的值为：%f",mid,func(mid));

return 0;

}

double func(double x){

double y;

y = pow(x,4.0) - 625\*pow(x,3.0) - 2\*x\*x + 1;

return y;

}

17.张一大答案：

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

#define SIZE 10

int main()

{

//两数组合并到新数组，然后新数组元素排序输出。

int len1,len2;

int a[SIZE];

int b[SIZE];

int merge[SIZE];

int i,j,k,temp;

printf("请输入第一个数组的个数：");

scanf("%d",&len1); //scanf自带换行错，而是自己打的回车。学习了。

printf("请输入第一个数组的元素：");

for(i=0;i<len1;i++){

scanf("%d",&a[i]);

}

printf("请输入第二个数组的个数：");

scanf("%d",&len2);

printf("请输入第二个数组的元素：");

for(i=0;i<len2;i++){

scanf("%d",&b[i]);

}

//合并数组

for(i=0;i<len1;i++){

merge[i] = a[i];

}

for(j=0;j<len2;j++){

merge[i++] = b[j];

}

//选择排序

for(i=0;i<len1+len2-1;i++){

k = i;

for(j=i+1;j<len1+len2;j++){

if(merge[k]>merge[j]){

k = j;

}

}

if(i!=k){

temp = merge[i];

merge[i] = merge[k];

merge[k] = temp;

}

}

//输出

for(i=0;i<len1+len2;i++){

printf("%-3d",merge[i]);

}

return 0;

}

1. 附录：

1. //以下数据存为class.txt

20074879 耿同学 男 江西

20077502 王同学 男 四川

20077508 肖同学 男 四川

20077524 王同学 男 四川

20077554 谢同学 男 四川

20067049 李同学 女 四川

20067054 王同学 男 四川

20067055 张同学 男 四川

20067061 刘同学 男 四川

20067049 杨同学 女 四川

20071020 赵同学 男 四川

20071080 周同学 女 四川

20071262 吴同学 女 四川

20072672 徐同学 女 四川

20072993 孙同学 女 黑龙江

20073728 朱同学 男 四川

20074864 马同学 女 北京

20074865 胡同学 女 甘肃

20074866 郭同学 女 广东

20074867 林同学 女 广西

20074868 何同学 女 陕西

20074869 高同学 女 四川

20074870 梁同学 女 四川

20074871 郑同学 女 四川

20074872 罗同学 女 四川

20074873 宋同学 女 四川

20074874 谢同学 男 福建

20074876 唐同学 男 黑龙江

20074877 韩同学 男 湖北

20074878 曹同学 男 湖南

20074879 许同学 男 江西

20074880 邓同学 男 辽宁

20074881 傅同学 男 宁夏

20074882 馮同学 男 山东

20074883 曾同学 男 四川

20074884 程同学 男 四川

20074885 蔡同学 男 四川

20074886 彭同学 男 四川

20074887 藩同学 男 四川

20074888 袁同学 男 四川

20074889 于同学 男 四川

20074890 董同学 男 四川

20074891 余同学 男 四川

20074892 苏同学 男 四川

20074893 叶同学 男 四川

20074894 吕同学 男 四川

20074895 魏同学 男 四川

20074896 蒋同学 男 四川

20074897 田同学 男 四川

20074898 杜同学 男 四川

20074899 丁同学 男 天津

20074900 沈同学 男 浙江

20075235 姜同学 男 四川

20075658 唐同学 男 四川

20076057 王同学 女 四川

20076060 穆同学 男 四川

20076098 陈同学 女 四川

20076101 王同学 女 四川

20076120 黄同学 女 四川

20077391 魏同学 男 四川

20077499 丁同学 男 四川

2. 输入样例：  
A.txt文件中的内容：  
24 24  
000000000000000000000000  
000000000011110000000000  
000000000011111000000000  
000000000111111000000000  
000000000111111000000000  
000000001111111100000000  
000000001110011100000000  
000000001110011110000000  
000000011110001110000000  
000000011100001110000000  
000000011100001111000000  
000000111100000111000000  
000000111000000111100000  
000001111000000111100000  
000001111000000011100000  
000001111111111111110000  
000011111111111111110000  
000011111111111111111000  
000111100000000001111000  
000111100000000000111000  
000111000000000000111100  
001111000000000000111100  
001111000000000000011110  
000000000000000000000000  
输出样例：  
B.txt中的内容：

24

10 4 10

10 5 9

9 6 9

9 6 9

8 8 8

8 3 2 3 8

8 3 2 4 7

7 4 3 3 7

7 3 4 3 7

7 3 4 4 6

6 4 5 3 6

6 3 6 4 5

5 4 6 4 5

5 4 7 3 5

5 15 4

4 16 4

4 17 3

3 4 10 4 3

3 4 11 3 3

3 3 12 4 2

2 4 12 4 2

2 4 13 4 1

24