

C 语言技能整理

/* 【程序填空】

函数 gcd() 计算整型数组 a 中相邻两元素的最大公约数，最小公倍数，
其中最大公约数存入数组 b 中，最小公倍数存入数组 c 中。

注意：请勿改动主函数 main() 中的其他内容。

-----*/

```
#include <stdio.h>
#include <conio.h>
void gcd(int a[],int n,int b[],int c[]){
    int i,x,y,z;
    /*******SPACE*****/
        for(i=0;i<n-1;i++){
            x=a[i];
            y=a[i+1];
            while(z=x%y){
                x=y;y=z;
            }
            b[i]=y;
    /*******SPACE*****/
            c[i]=a[i]*a[i+1]/b[i];
        }
    }
int main(){
    int a[5]={8,12,5,7,14};
    int b[4],c[4];
    int i;
    gcd(a,5,b,c);
    for(i=0;i<5;i++)
        printf("%d\t",a[i]);
    printf("\n");
    for(i=0;i<4;i++)
        printf("%d\t",b[i]);
    printf("\n");
    for(i=0;i<4;i++)
        printf("%d\t",c[i]);
    printf("\n");
    getch();
    return 0;
}
/*-----
```

【程序改错】

主函数 main()调用函数 change()将二维数组 strow 中的字符按列存入一维数组 strcol 中。

```
-----*/
#include <stdio.h>
#include <conio.h>
/*****FOUND*****/
void change(char t[][5],char s[]){
    int i,j;
    for(j=0;j<5;j++)
        for(i=0;i<3;i++)
/*****FOUND*****/
            s[3*j+i]=t[i][j];
/*****FOUND*****/
    s[3*j+i-3]='\0';/*标准答案 s[3*j]=0*/
}

int main(){
    char strow[][5]={ {'C','r','r',' ','s'},{' ','o','a','T','t'},{'P','g','m','e','!'} };
    char strcol[16];
    change(strow,strcol);
    puts(strcol);
    getch();
    return 0;
}
/*-----
```

【程序设计】

假设英文大小写字母的权值分别为:

A-E,a-e	1;
F-J,f-j	2;
K-O,k-o	3;
P-T,p-t	4;
U-Y,u-y	5;
Z,z	6

计算主函数 main()中数组 words 中的各英文单词的权重(英文单词的权重为该单词各字母权值之和),并按权重进行降序排序。

例如,单词"World"的权重为 16(W 为 5;o 为 3;r 为 4;l 为 3;d 为 1,5+3+4+3+1=16)。

编写程序:

1.编写函数 void calculate(char w[][20],int n,int p[]), 计算数组 w 中 n 个英文单词的权重, 并

将权重存放在数组 p 中。

2.编写函数 void sort(int p[],int n,char w[][20]), 对数组 p 中 n 个单词的权重进行降序排序, 权重对应单词在数组 w 中的位置也要作相应调整。

```
-----*/

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>

void calculate(char w[][20],int n,int p[]){
/*****Program*****/
    int i,j,s;
    for(i=0;i<n;i++)
    {
        j=0;s=0;
        while(w[i][j])
            s=s+(tolower(w[i][j++])-65)/5+1;
        p[i]=s;
    }
/*****End*****/
}

void sort(int p[],int n,char w[][20]){
/*****Program*****/
    int i,j,t,max;
    char k[20];
    for(i=0;i<n;i++)
    {
        max=i;
        for(j=i+1;j<n;j++)
            if(p[j]>p[max])
                max=j;
        t=p[max];
        p[max]=p[i];
        p[i]=t;
        strcpy(k,w[max]);
        strcpy(w[max],w[i]);
    }
}
```

```

        strcpy(w[i],k);
    }
/*****End*****/
}

int main(){
    char words[5][20]={ {"JiangSu"}, {"Teachers"}, {"University"}, {"of"}, {"Technology"} };
    int value[5]={0};
    int i;
    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL){
        printf("File open error\n");
        exit(0);
    }
    calculate(words,5,value);
    for(i=0;i<5;i++){
        fprintf(fp,"%-20s%4d\n",words[i],value[i]);
        printf("%-20s%4d\n",words[i],value[i]);
    }
    printf("\n");
    fprintf(fp,"\n");
    sort(value,5,words);
    for(i=0;i<5;i++){
        fprintf(fp,"%-20s%4d\n",words[i],value[i]);
        printf("%-20s%4d\n",words[i],value[i]);
    }
    fclose(fp);
    getch();
    return 0;
}

```

/*-----

【程序填空】

函数 reverse()对字符串 str 进行逆序。

注意：请勿改动主函数 main()中的其他内容。

-----*/

```

#include<stdio.h>
#include<string.h>
#include<conio.h>

```

```

void reverse(char str[]){
    int    len, j;

```

```

        char c;
        /**SPACE***/
        len =strlen(str);
        /**SPACE***/
        for (j=len-1;j>len/2; j--){/* for (j=len-1;j>=len/2; j--)*
            c=str[j];
            str[j]=str[len-1-j];
            str[len-1-j]=c;
        }
    }
int main(){
    char a[80];
    printf("Please enter a string: ");
    gets(a);
    reverse(a);
    printf("The inversed string is: ");
    puts(a);
    getch();
    return 0;
}
/*-----

```

【程序改错】

在主函数中 main()中输入年、月、日，然后计算该天是这一年的第几天。其中函数 LeapYear()是判断某年是否为闰年。

```

-----*/

#include <stdio.h>
#include <conio.h>
int LeapYear(int year){
    /**SPACE***/
    return ((year%4==0 && year%100) || (year%400==0));
}
int main(){
    int
mon_days[2][11]={ {31,28,31,30,31,30,31,31,30,31,30}, {31,29,31,30,31,30,31,31,30,31,30}};
    int year,mon,day,days;
    int i;
    scanf("%d %d %d",&year,&mon,&day);
    days=day;
    /**SPACE***/
    for(i=0;i<mon-1;i++)
    /**SPACE***/

```

```

        days=days+mon_days[LeapYear(year)][i];
    printf("%d\n",days);
    getch();
    return 0;
}

```

```

/*-----

```

【程序设计】

主函数 main()中 str_b 是这样的字符串：若干个长度不等的，连续的'0'、'1'字符组成的字符串被长度不等的'.'字符间隔。将连续的'0'、'1'字符子串看成是二进制数。要求将其转换为十进制整数。其中字符子串的首字符代表二进制数的符号位，'0'表示正数，'1'表示负数(例如："0111"表示 7 , "1111"表示-7)。

编写程序：

- 1.编写函数 int convert(char b[],int d[])，将字符数组 b 中连续二进制数字字符串转换为十进制整数，并存入整型数组 d 中。函数返回十进制整数的个数。
- 2.编写函数 void sort(int d[],int n)，对数组 d 中 n 个元素进行升序排序。

```

-----*/

```

```

#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
int convert(char b[],int d[]){
    /*******Program*****/
    int i=0,n=0,s=0,flagks=0,flagzf=1;
    while(b[i])
    {
        if(b[i]!='.')
            if(!flagks)
            {
                i++;
                continue;
            }
        else
            {

```

```

        d[n++]=s*flagzf;
        flagks=0;
        s=0;
        i++;
        continue;
    }
else
    if(!flagks)
    {
        flagks=1;
        if(b[i]!=48)
            flagzf=-1;
        else
            flagzf=1;
        i++;
        continue;
    }
else
    {
        s=s*2+(b[i]-48);
        i++;
        continue;
    }

}

return n;
/*****End*****/
}

void sort(int d[],int n){
/*****Program*****/
    int i,j,t;
    for(i=0;i<n-1;i++)
        for(j=i+1;j<n-1;j++)
            if(d[i]>d[j])
            {
                t=d[i];
                d[i]=d[j];
                d[j]=t;
            }

```

```

/*****End*****/
}

int main(){
    char str_b[100]="...111100.01111.01100111...0111..110000.011..";
    int int_d[10];
    int i,k;
    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL){
        printf("File open error\n");
        exit(0);
    }
    k=convert(str_b,int_d);
    for(i=0;i<k;i++){
        printf("%d\t",int_d[i]);
        fprintf(fp,"%d\t",int_d[i]);
    }
    printf("\n");
    fprintf(fp,"\n");
    sort(int_d,k);
    for(i=0;i<k;i++){
        printf("%d\t",int_d[i]);
        fprintf(fp,"%d\t",int_d[i]);
    }
    printf("\n");
    fprintf(fp,"\n");
    fclose(fp);
    getch();
    return 0;
}
/*-----

```

【程序改错】

程序去除主函数 main()中字符串 str 中的非字母字符。
 函数 IsAlpha()的功能是判断字符 c 是否为字母。

```

-----*/

#include <stdio.h>
#include <conio.h>
int IsAlpha(char c){
/*****FOUND*****/
    return ((c>='a' && c<='z')||(c>='A' && c<='Z'));
}

```



```

}
int main(){
    char str[]={ "/* One World,One Dream! */" };
    int prev,next;
    prev=next=0;
    while(str[next]){
        if(IsAlpha(str[next])){
/*****FOUND*****/
            str[prev]=str[next];
            prev++;
        }
        next++;
    }
/*****FOUND*****/
    str[prev]=0;
    puts(str);
    getch();
    return 0;
}
/*-----

```

【程序设计】

主函数 main()中一维数组 ring[9]存放数字 1-9。将 ring 看成是一个首尾相接的环。将 9 个数分成 3 段，第 1 段为 1 个 2 位数，第 2 段为 1 个 3 位数，第 3 段为 1 个 4 位数，程序计算这 3 段数之和。要求从环的第 1 个数开始，直到第 9 个数，依上述规则进行处理。同时在这些和中寻找 77 的整数倍的数。

例如：从环的第一个数开始的 3 段数为 12,345,6789,其和为 7146；从环的第二个数开始的 3 段数为 23,456,7891,其和为 8370；

.....

编写程序：

- 1.编写函数 void calculate(int ring[],int st[]),从数组 ring 的第 1 个数开始,将 9 个数分成 3 段,第 1 段为 1 个 2 位数,第 2 段为 1 个 3 位数,第 3 段为 1 个 4 位数,程序计算所有 3 段数之和。并将所有 3 段数之和存入数组 st 中。
- 2.编写函数 int check(int st[],int t77[]),在数组 st 中寻找 77 的整数倍的数,存入数组 t77 中。函数返回其个数。

```

-----*/

```

```

#include<stdio.h>

```

```

#include<stdlib.h>
#include<conio.h>
void calculate(int ring[],int st[]){
/*****Program*****/
int i,n=0;
int a,b,c;
for(i=0;i<9;i++)
{
    a=ring[i]*10+ring[(i+1)%9];
    b=ring[(i+2)%9]*100+ring[(i+3)%9]*10+ring[(i+4)%9];
    c=ring[(i+5)%9]*1000+ring[(i+6)%9]*100+ring[(i+7)%9]*10+ring[(i+8)%9];
    st[n++]=a+b+c;
}

/****End*****/
}
int check(int st[],int t77[]){
/*****Program*****/
int n=0,i;
for(i=0;i<9;i++)
    if(st[i]%77==0)    t77[n++]=st[i];
return n;
/****End*****/
}
int main(){
    int ring[9]={ 1,2,3,4,5,6,7,8,9};
    int st[9],t77[9],i,k;
    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL){
        printf("File open error\n");
        exit(0);
    }
    calculate(ring,st);
    for(i=0;i<9;i++){
        printf("%d\t",st[i]);
        fprintf(fp,"%d\t",st[i]);
    }
    printf("\n");
    fprintf(fp,"\n");
    k=check(st,t77);
    for(i=0;i<k;i++){
        printf("%d\t",t77[i]);

```

```

        fprintf(fp,"%d\t",t77[i]);
    }
    printf("\n");
    fprintf(fp,"\n");
    fclose(fp);
    getch();
    return 0;
}

/*-----

```

【程序填空】

在一维数组 a 中找出最大，最小元素，并将最大元素和数组最后一个元素交换，最小元素和数组首元素交换。

注意：请勿改动主函数 main() 中的其他内容。

```

-----*/

#include <stdio.h>
#include <conio.h>
int main(){
    int a[10]={ 15,8,12,6,10,1,4,5,9,-3};
    int i,t,max_index,min_index;
    max_index=min_index=0;
    for(i=1;i<10;i++){
        if(a[i]>a[max_index])
            max_index=i;
        if(a[i]<a[min_index])
            min_index=i;
    }
    t=a[0];a[0]=a[min_index];a[min_index]=t;
    /******SPACE*****/
    if(max_index==0){
        t=a[min_index];a[min_index]=a[9];a[9]=t;
    }
    else{
        /******SPACE*****/
        t=a[max_index];a[max_index]=a[9];a[9]=t;
    }
    for(i=0;i<10;i++)
        printf("%d\t",a[i]);
    printf("\n");

    return 0;
}

```

```
}
```

```
/*-----
```

【程序改错】

主函数 main()统计字符串 str 中的各字母(不区分大小写)及其他字符出现的次数，并显示统计结果(次数为 0，则不显示)。

```
-----*/
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#include <ctype.h>
```

```
int main(){
```

```
    char str[]={ "---JiangSu Teachers University of Technology---"};
```

```
    int chnum[27]={0},i=-1;
```

```
    /******FOUND******/
```

```
    while(str[++i]){
```

```
        if(isalpha(str[i]))
```

```
    /******FOUND******/
```

```
        chnum[toupper(str[i])-'A']++;
```

```
    else
```

```
        chnum[26]++;
```

```
    }
```

```
    for(i=0;i<27;i++)
```

```
    /******FOUND******/
```

```
        if(chnum[i]==0)
```

```
            if(i==26)
```

```
                printf("other ch=%d\n",chnum[26]);
```

```
            else
```

```
                printf("%c(%c)=%d\n",i+65,i+97,chnum[i]);
```

```
    return 0;
```

```
}
```

```
/*-----
```

【程序设计】

将主函数 main()中字符数组 IPb 中的 32 个'0','1'字符看做一个 32 位的二进制形式的 IP 地址，将其转换为 4 个十进制数整数形式的 IP 地址。并以点分十进制记法(XXX.XXX.XXX.XXX)输出 IP 地址。

同时指出该地址所属的地址类。IP 地址类见下面分类：

A 类地址：1.0.0.1-126.255.255.254

B 类地址：128.0.0.1-191.255.255.254

C 类地址：192.0.0.1-223.255.255.254

D 类地址：224.0.0.0-239.255.255.255

E 类地址：240-

编写程序：

1.编写函数 void btod(char ipb[],int ipd[]),将字符数组 ipb 中由字符组成的二进制形式的 IP 地址转换为十进制整数形式的 IP 地址，存入数组

ipd 中。

2.编写函数 char classto(int ip), 依据参数 ip 的值，判断 IP 所属的地址类，函数返回所属类（返回字符'A'~'E'）。

```
-----*/

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

void btod(char ipb[],int ipd[]){
/*****Program*****/
int i,s=0,n=0;
for(i=0;i<33;i++)
{
    s=s*2+ipb[i]-'0';
    if((i+1)%8==0)
    {
        ipd[n++]=s;
        s=0;
    }
}

/*****End*****/
}

char classto(int ipd){
/*****Program*****/
    if(ipd<128) return 'A';
    else if(ipd<192) return 'B';
        else if(ipd<224) return 'C';
            else if(ipd<240) return 'D';
                else return 'E';
}
```

```

/*****End*****/
}

int main(){
    char IPb[33]="11000000111011001110001110000011",c;
    int IPd[4]={0},i;
    FILE *fp;
    btod(IPb,IPd);
    c=classto(IPd[0]);
    if((fp=fopen("DATA.TXT","w"))==NULL){
        printf("File open error\n");
        exit(0);
    }
    for(i=0;i<3;i++){
        printf("%d.",IPd[i]);
        fprintf(fp,"%d.",IPd[i]);
    }
    printf("%d\t",IPd[3]);
    fprintf(fp,"%d\t",IPd[3]);
    printf("%c 类地址\n",c);
    fprintf(fp,"%c 类地址\n",c);
    fclose(fp);

    return 0;
}
/*-----

```

【程序填空】

题目：对只包含'-'和字母字符的字符串 str 进行如下处理：
 删除 str 中前导和尾部连续的'-'字符，保留串中字母字符以及字母间的'-'字符。
 例如，str 中初始字符串为"---C--Language--Test---"。
 处理后，str 中的字符串为"C--Language--Test"。
 程序中 isalpha 函数功能是判断参数是否是字母，是字母，函数值为真，否则函数值为假。

注意：请勿改动主函数 main()中的其他内容。

```

-----*/

#include <stdio.h>
#include <ctype.h>
void main(){

```

```

char str[]={"---C--Language--Test---"};
int i,start,end;
i=0;
while(str[i++]=='-') ;
start=i-1;
/*****SPACE*****/
while(str[i] != '\0'){
    if(isalpha(str[i]))
        end=-1;
    else if(isalpha(str[i-1]))
        end=i-1;
    i++;
}
str[end+1]='\0';
for(i=start;i<=end+1;i++)
/*****SPACE*****/
    str[i-start] = str[i];
printf("%s\n",str);
}
/*-----

```

【程序改错】

题目：函数 unsigned fun(long s, int w) 返回长整型变量 s 中第 w 位的值，例如调用 fun(216783,1) 函数返回 3，调用 fun(216783,3) 函数返回 7。main() 从键盘接收一个长整数给变量 s，通过调用 fun() 函数求 s 奇数位之和，例如数 216783 各奇数位数字为 1，7 和 3，它们的和为 11。

下面给定的程序存在错误，请改正。

注意：不得增行或删行，也不得更改程序的结构。

```

-----*/

#include <stdio.h>
unsigned fun(long s, int w)
{
    int d;
    while(w>0)
    {
/*****FOUND*****/
        d=s%10;
        s /=10;
    }
}

```

```

        w--;
    }
    return d;
}

main()
{
    /*******FOUND*****/
    unsigned tot=0, w=1;
    long s, step;
    printf(" Enter a long integer number:");
    scanf("%ld", &s);
    step = s;
    while (step!=0)
    {
        /*******FOUND*****/
        tot = tot+fun( s ,w);
        w+=2;
        step /=100;
    }
    printf("tot=%d\n", tot);
    return 0;
}
/*-----

```

【程序设计】

题目：找出二维整型数组 a[4][4]中每行元素的最大值，并将该元素移动至该行首列。

编写程序：

- 1、编写 void findmax(int a[][N], int n[])函数：找出二维数组 a 中每行元素的最大值，并将该元素的列下标存入数组 n 中，即第 0 行元素最大值的列下标存入 n[0]，...，第 3 行元素最大值的列下标存入 n[3]。
- 2、编写 void move(int a[][N],int n[])函数：根据数组 n 中保存的数据，将 a 数组中每行最大值元素移动到该行首列，要求移动后该元素与该行其他元素的相对位置不变。

例如：

a 数组的初始数据为：

-2 13 6 9

25	4	0	11
-8	3	10	16
4	7	5	20

经过处理后 a 数组中数据为:

13	6	9	-2
25	4	0	11
16	-8	3	10
20	4	7	5

注意：请勿改动主函数 main() 中的任何语句。

-----*/

```
#include <stdio.h>
#include <stdlib.h>
#define N 4
void findmax(int a[][N], int n[]){
    /******Program*****/
    int i,j,max;
    for(i=0;i<N;i++)
    {
        max=0;
        for(j=1;j<N;j++)
            if(a[i][j]>a[i][max])
                max=j;
        n[i]=max;
    }
    /****** End *****/
}

void move(int a[][N],int n[]){
    /******Program*****/
    int i,j,t,k;
    for(i=0;i<N;i++)
    {
        k=n[i];
        while(k)
        {
            t=a[i][0];
            for(j=0;j<N-1;j++)
                a[i][j]=a[i][j+1];
            a[i][j]=t;
            k--;
        }
    }
}
```

```

    }

/*****      End      *****/
}

void main(){
    int i,j;
    int a[N][N]={ {-2,13,6,9},{25,4,0,11},{-8,3,10,16},{4,7,5,20}},n[4];
    FILE *fp;
    findmax(a,n);
    move(a,n);
    if((fp=fopen("data.txt","w"))==NULL){
        printf("open file error\n");
        exit(0);
    }
    for(i=0;i<N;i++){
        for(j=0;j<N;j++){
            fprintf(fp,"%d\t",a[i][j]);
            printf("%d\t",a[i][j]);
        }
        fprintf(fp,"\n");
        printf("\n");
    }
    fclose(fp);
}
/*-----

```

【程序填空】

题目：下面的程序首先随机产生 20 个大于或等于 900 且小于 1000 的素数，然后将这些素数保存在数组 array 中，并将这些素数
在屏幕上以每行打印 5 个的格式输出，请将程序补充完整。

注意：请勿改动主函数 main() 中的其他内容
且不能使用 C 语言的库函数。

```

-----*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/*函数 isPrime(int m)用于判断整数 m 是否为素数*/
int isPrime(int m)

```

```

{
    int flag=0;
    int i=2,k;
    k=(int)sqrt(m);
    /*******SPACE*****/
    while (i<=k && m%i) i++;
    if(i>k) flag=1;
    return flag;
}

void main()
{
    int array[20];
    int i,t;
    i=0;
    while(i<20)
    {
        t=rand()%100+900;
        /*******SPACE*****/
        if (isPrime(t))
        {
            array[i]=t;
            printf("%6d",t);
            i++;
            if (!(i%5)) printf("\n");
        }
    }
    printf("\n");
}
/*-----

```

【程序改错】

题目：主函数 main()中调用 count()函数统计子串 substr 在字符串 str 中出现的次数。例如字符串 str 为” abdr gabdd”，子串 substr 为” ab”，则子串在该字符串出现的次数为 2。

下面给定的程序存在错误，请改正。

注意：不得增行或删行，也不得更改程序的结构。

-----*/

#include<stdio.h>

```

int count(char str[],char substr[])
{
    /*******FOUND*****/
    int i,j,k,num=0;
    for(i=0;str[i]!='\0';i++)
    {
        /*******FOUND*****/
        for(j=i,k=0;substr[k]==str[j];k++,j++)
            if(substr[k+1]!='\0')
                { num++;break; }
    }
    return(num);
}

int main()
{
    char str[80],substr[80];
    int num=0;
    printf("Please input string:");
    gets(str);
    printf("Please input substring:");
    gets(substr);
    /*******FOUND*****/
    num=count(str,substr);
    if(num)
        printf("Substing :%d\n",num);
    else
        printf("%s is not substring!\n",substr);
    return 0;
}
/*-----

```

【程序设计】

题目：主函数 main()中一维数组 a 为测试数据,程序将测试数据进行加密，加密后的数据存入数组 b 中，判断数组 b 中是否存在素数，将所有素数存入数组 c 中，并输出。

编写程序：

1. 编写函数 void encode(int x[],int n,int y[]),将数组 x 中 n 个不超过 3 位的正整数逐个做加密处理，并将加密后的数据存入数组 y 中。加密方法如下：将正整数的每一位用该位数字加该位序号的值替换（序号规定：个位为 3，十位为 2，百位为 1），若结果大于 9，

则用该数除以 10 的余数替换。

2. 编写函数 `int CheckPrime(int y[],int n,int c[])`，
判断数组 `y` 中 `n` 个整数是否是素数，并将所有素数存入数组 `c` 中，函数返回素数个数。

例如：

测试数据：

25,90,89,24,125

运行结果：

c[0]=13 is prime

c[1]=2 is prime

c[2]=47 is prime

注意：请勿改动主函数 `main()` 中的任何语句。

```
-----*/

#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<math.h>

void encode(int x[],int n,int y[])
{
    /*******Program*****/
    int i,gw,sw,bw;
    for(i=0;i<n;i++)
    {
        gw=(x[i]%10+3)%10;
        sw=((x[i]%100)/10+2)%10;
        if(x[i]/100)
            bw=(x[i]/100+1)%10;
        else
            bw=0;
        y[i]=gw+sw*10+bw*100;
    }
    /******* End *****/
}

int CheckPrime(int y[],int n,int c[])
{
    /*******Program*****/
    int i,j,flag,k=0;
    for(i=0;i<n;i++)
```

```

    {
        flag=1;
        for(j=2;j<=sqrt(y[i]);j++)
            if(y[i]%j==0)
                flag=0;
        if(flag)
            c[k++]=y[i];
    }
return k;
/*****      End      *****/
}

int main()
{
    int i,a[10]={25,90,89,24,125},b[10]={0},c[10]={0};
    int num=0,n;
    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("File open error\n");
        exit(0);
    }

    n=5;
    encode(a,n,b);
    num=CheckPrime(b,n,c);

    for(i=0;i<num;i++)
    {
        printf("c[%d]=%d is prime\n",i,c[i]);
        fprintf(fp,"c[%d]=%d is prime\n",i,c[i]);
    }
    fclose(fp);

    return 0;
}
/*-----

```

【程序填空】

题目：随机产生 10 个范围在[0,99]之间，互不相同的正整数存入数组 t，并求出这 10 个数据的最小值和次小值。其中最小值存入 min1，次小值存入 min2。程序中 rand()函数功能是产生一个随机正整数。

注意：请勿改动主函数 main() 中的其他内容。

```
-----*/

#include <stdio.h>
#include <stdlib.h>
void main(){
    int n[100]={0},t[10];
    int i=0,k;
    int min1=100, min2=100;
    srand(32767);
    while(i<10){
        do{
            k=rand()%100;
        }while(n[k]);
        n[k]=1;
        /******SPACE*****/
        t[i++]=k;
    }
    for(i=0;i<10;i++){
        printf("%d\t",t[i]);
        if(t[i]<min1){
            /******SPACE*****/
            min2=min1;
            min1=t[i];
        }
        else if(t[i]<min2)
            min2=t[i];
    }
    printf("\nmin1= %d,min2= %d\n",min1,min2);
}
/*-----

【程序改错】
-----
```

题目：函数 main() 首先从键盘接收 10 个整数，依次存放在 a 数组中；其次，调用 fun() 函数在这 10 个整数中找出第一个奇数，并返回该数的下标值，如果在这 10 个数中无奇数，则返回 -1；第三，如果 fun() 函数返回值不等于 -1 则把找到的数与第 1 个数对换位置。

下面给定的程序存在错误，请改正。

注意：不得增行或删行，也不得更改程序的结构。

-----*/

```
#include<stdio.h>
#define N 20
int fun(int a[])
{
    int i;
    for (i=0; i<10; i++)
    {
        /******FOUND******/
        if (a[i]%2==1)
            break;
    }
    /******FOUND******/
    if (i<10) return i;
    else return -1;
}

main()
{
    int a[N], i, r, step;
    for (i=0; i<10; i++)
    {
        /******FOUND******/
        scanf("%d",&a[i]);
    }
    r = fun(a);
    if (r!=-1)
    {
        step = a[r];
        a[r] = a[0];
        a[0] = step;
    }
    for(i=0;i<10;i++)
        printf("%d ",a[i]);
    printf("\n");
    return 0;
}
/*-----
```

【程序设计】

题目：程序对一维数组 a 中数据进行升序排序后，对数组 a 中重复出现的数据，删除重复数据，保留一个。对只出现一次的数据则全部保留。

编写程序：

- 1、编写 sort(int a[],int n)函数，对数组 a 中 n 个元素进行升序排序。
- 2、编写 int delelem(int a[],int n)函数，对数组 a 中 n 个数据删除重复数据，只保留一个。对只出现一次的数据则全部保留。函数返回数组中剩余元素的个数。

例如：

数组 a 的初始数据为： 2,5,2,4,20,0,4,-5,-5,16,2,-4,20。

升序排序后为： -5,-5,-4,0,2,2,2,4,4,5,16,20,20。

删除重复数据后为： -5,-4,0,2,4,5,16,20。

注意：请勿改动主函数 main()中的任何语句。

-----*/

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
void sort(int a[],int n){
```

```
/******Program*****/
```

```
int i,j,min,t;
```

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

```
{
```

```
    min=i;
```

```
    for(j=i+1;j<n;j++)
```

```
        if(a[j]<a[min])
```

```
            min=j;
```

```
    t=a[i];
```

```
    a[i]=a[min];
```

```
    a[min]=t;
```

```
}
```

```
/****** End *****/
```

```
}
```

```
int delelem(int a[],int n){
```

```
/******Program*****/
```

```

int i,j;
for(i=0;i<n-1;i++)
{
    if(a[i]==a[i+1])
    {
        for(j=i+1;j<n;j++)
            a[j]=a[j+1];

        n--;
    }
    else
        continue;
}
/*****      End      *****/
}

void main(){
    int a[]={ 12,2,1,-3,7,-3,4,15,2,0,1,15};
    int n,i;
    FILE *fp;
    n=sizeof(a)/sizeof(int);
    sort(a,n);
    for(i=0;i<n;i++)
        printf("%d\t",a[i]);
    printf("\n");
    n=delelem(a,n);
    if((fp=fopen("data.txt","w"))==NULL){
        printf("File open error\n");
        exit(0);
    }
    for(i=0;i<n;i++){
        fprintf(fp,"%d\t",a[i]);
        printf("%d\t",a[i]);
    }
    printf("\n");
    fclose(fp);
}
/*-----

```

【程序填空】

题目：将输入的一个十进制正整数转换为十六进制字符形式并显示。

注意：请勿改动主函数 main() 中的其他内容。

```
-----*/

#include <stdio.h>
void main(){
    char dex[17]={"0123456789ABCDEF"};
    int  c[8],i,base=16;
    int n;
    printf("输入十进制正整数 ");
    scanf("%d",&n);
    i=0;
    do{
        /*****SPACE*****/
        c[i++]=n%base;
        n=n/base;
    }while(n);
    printf("对应的十六进制数 ");
    for(--i;i>=0;i--)
        /*****SPACE*****/
        printf("%c",dex[c[i]]);
    printf("\n");
}
/*-----
```

【程序改错】

题目：函数 main() 首先从键盘接收一串字符，以回车符结束串输入，并保存在变量 str1 中，输入一个字符给变量 ch；其次，调用 fun() 函数将字符串 str1 中的所有为 ch 字符去掉；最后输出字符串 str1。

下面给定的程序存在错误，请改正。

```
-----
注意：不得增行或删行，也不得更改程序的结构。
-----*/
```

```
#include <stdio.h>
void fun(char str[], char ch)
{
    int i=0, j=0;
    while (str[i]!=0)
    {
        if (str[i] != ch)
```

```

        {
/*****FOUND*****/
                str[j++]=str[i];
        }
        i++;
    }
/*****FOUND*****/
    str[j]='\0';
}

main()
{
    char str1[200], ch;
    int i=0;
/*****FOUND*****/
    while((str1[i]=getchar())!='\n')
        i++;
    str1[i] = '\0';
    scanf("%c",&ch);
    fun(str1,ch);
    printf("%s\n",str1);
    return 0;
}
/*-----

```

【程序设计】

题目：若一个数和该数的逆序数相同，就称该数为回文数。对于一个非回文数，加上该数的逆序数后，其和可能是回文数。如和仍然是非回文数，则有限次重复上述步骤总能得到回文数。

例如，87 是非回文数，则

87+78=165	165 非回文数
165+561=726	726 非回文数
726+627=1353	1353 非回文数
1353+3531=4884	4884 回文数

主函数 main()从文件中读入一个正整数，判断其是否是回文数。如果该数是回文，则输出该数，如果不是回文，则使用上述算法，求出回文数。

编写程序：

- 1、编写 int reverse(int x)函数：求参数 x 的逆序数，并将

该逆序数返回。

2、编写 int ispalin(int x)函数：判断参数 x 是否是回文数，是回文数，函数返回真，否则函数返回假。

3、编写 int palin(int x)函数：如果参数 x 是回文数，函数返回 x；如果 x 不是回文数，则根据上述算法求出回文数后，返回该回文数。

注意：请勿改动主函数 main()中的任何语句。

-----*/

```
#include <stdio.h>
#include <stdlib.h>
int reverse(int x){
    /*******Program*****/
    long s=0;
    while(x)
    {
        s=s*10+x%10;
        x=x/10;
    }
    return s;
    /******* End *****/
}
```

```
int ispalin(int x){
    /*******Program*****/
    int i=0,j=0,flag=1;
    int sj[100];
    while(x)
    {
        sj[j++]=x%10;
        x=x/10;
    }
    for(;i<=j/2;i++)
    {
        if(sj[i]!=sj[j-1-i])
        {
            flag=0;
            break;
        }
    }
}
```

```

return flag;
/*****      End      *****/
}

int palin(int x){
/*****Program*****/
if(ispalin(x))
    return x;
else
    return palin(x+reverse(x));
/*****      End      *****/
}

void main(){
    int  x, y;
    FILE *fpin,*fpout;
    if((fpin=fopen("input.txt","r"))==NULL){
        printf("Input file open error\n");
        exit(0);
    }
    fscanf(fpin,"%d",&x);
    fclose(fpin);
    y=palin(x);
    printf("initial is %d\n",x);
    printf("palindrome is %d\n",y);
    if((fpout=fopen("data.txt","w"))==NULL){
        printf("Output file open error\n");
        exit(0);
    }
    fprintf(fpout,"initial is %d\n",x);
    fprintf(fpout,"palindrome is %d\n",y);
    fclose(fpout);
}
/*-----

```

【程序填空】

题目：统计字符串 `str` 中单词的数量、最长单词的字母数、最短单词字母数。单词之间至少有一个空格。假设 `str` 中最长单词不超过 20 个字母。程序中 `isalpha` 函数功能是判断参数是否是字母字符，如果是字母字符，函数返回真，否则函数返回假。

注意：请勿改动主函数 main() 中的其他内容。

```
-----*/

#include <stdio.h>
#include <ctype.h>
void main(){
    int i,len=0;
    int numofword,maxlen,minlen;
    char str[1000]="Robots can feel happy, sad,   angry, drowsy,   hungry or fearful.   They also
have various personalities.";
    /*****SPACE*****/
    numofword=0;
    maxlen=0;
    minlen=20;
    for(i=0;str[i];i++){
        if(!(isalpha(str[i])) && isalpha(str[i-1]))){
            maxlen=len>maxlen? len: maxlen;
            minlen=len<minlen? len: minlen;
            len=0;
            numofword++;
        }
        else if(isalpha(str[i]))
    /*****SPACE*****/
        len++;
    }
    printf(" 单 词 数  =%d, 最 长 单 词 字 母 数  =%d, 最 短 单 词 字 母 数
=%d\n",numofword,maxlen,minlen);
}
/*-----
```

【程序改错】

题目：N 个有序整数数列已放在一维数组 a 中，下列给定程序中，
函数 fun 的功能是：利用折半查找算法找整数 m 在数组中的
位置。若找到，则返回其下标值；反之，则返回 -1。

下面给定的程序存在错误，请改正。

注意：不得增行或删行，也不得更改程序的结构。

```
-----*/

#include <stdio.h>
#define N 10
```

```

int fun(int a[],int m)
{
    /*******FOUND*****/
    int left=0, right=N-1, mid;
    while(left<=right)
    {
        mid=(left+right)/2;
        if (m>a[mid])
            /*******FOUND*****/
            left =mid +1;
        else if (m<a[mid])
            /*******FOUND*****/
            right= mid -1;
        else return (mid);
    }
    return(-1);
}

main()
{
    int i, a[N]={-3,4,7,9,13,45,67,89,100,180}, k, m;
    printf("a 数组中的数据如下： ");
    for(i=0;i<N;i++)
        printf("%d ",a[i]);
    printf("\nEnter m:");
    scanf("%d",&m);
    k=fun(a,m);
    if (k>=0)
        printf("m=%d,index=%d\n",m,k);
    else
        printf("Not be found!\n");
    return 0;
}
/*-----

```

【程序设计】

题目：主函数 main()将字符串 a 中若干个连续的数字字符串转换成整数（不考虑正负符号），保存在数组 n 中，并对数组 n 中元素进行升序排序，最后计算最小值、次小值的最大公约数。

系统函数 int isdigit(char c) 的功能是判断参数 c 是否是数字字符。如果 c 是数字字符，函数返回真，否则返回假。

编写程序：

- 1、编写 int abstract(char a[],int n[])函数：将字符串 a 中的连续数字字符转换为整数，保存在数组 n 中,函数返回整数个数。
- 2、编写 void sort(int n[],int k)函数：对数组 n 中的 k 个元素进行升序排序。
- 3、编写 int gcd(int a,int b)函数：求参数 a,b 的最大公约数。

注意：请勿改动主函数 main()中的任何语句。

```
-----*/

#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
int abstract(char a[],int n[]){
/*****Program*****/
int k=0,i=0,flag=0,s=0;
while(a[i])
{ if(isdigit(a[i]))
    flag=1;
  else
  {
    i++;
    continue;
  }
  if(isdigit(a[i])&&!isdigit(a[i-1]))&&flag)
  {
    s=0;
    s=s*10+a[i++]-'0';
    continue;
  }
  else
    if(isdigit(a[i])&&isdigit(a[i+1]))&&flag)
      s=s*10+a[i++]-'0';
    else
      if(!isdigit(a[i+1]))&&flag)
      {
        s=s*10+a[i++]-'0';
        n[k++]=s;
      }
  }
}
```

```

return k;
/*****      End      *****/
}

void sort(int n[],int k){
/*****Program*****/
int i,j,min,t;
for(i=0;i<k;i++)
{
    min=i;
    for(j=i+1;j<k;j++)
        if(n[j]<n[min])
            min=j;
    t=n[i];
    n[i]=n[min];
    n[min]=t;
}
/*****      End      *****/
}

int gcd(int a,int b){
/*****Program*****/
int i,t;
if(a<b)
{
    t=a;
    a=b;
    b=t;
}
for(i=b;i>=1;i--)
    if(a%i==0&& b%i==0)
        return i;
return 1; /*可以省略*/
/*****      End      *****/
}

void main(){
    char a[100]={ "kn--23,q,37.&#% % 18'w94jAF% 130Enter12--+55" };
    int  n[10],k,i;
    FILE *fp;
    k=abstract(a,n);
    sort(n,k);
    if((fp=fopen("data.txt","w"))==NULL){
        printf("File open error\n");

```

```

        exit(0);
    }
    for(i=0;i<k;i++){
        fprintf(fp,"%d\t",n[i]);
        printf("%d\t",n[i]);
    }
    k=gcd(n[0],n[1]);
    fprintf(fp,"\n%d, %d 's gcd=%d\n",n[0],n[1],k);
    printf("\n%d, %d 's gcd=%d\n",n[0],n[1],k);
    fclose(fp);
}
/*-----

```

【程序填空】

题目：程序首先将随机产生的 12 个水仙花数存放于一个 3 行 4 列的数组 array 中，然后找出其中的最大数所在的行号与列号，并将行号与列号输出在屏幕上。

说明：所谓水仙花数是指其各位数字立方和等于该数本身的数。

注意：除要求填空的位置之外，请勿改动程序中的其他内容。
且不能使用 C 语言的库函数

-----*/

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

```

```

int narcissus()
{
    int M,N,t,sum,find=0;

    while(!find)
    {
        sum=0;
        N=rand()%900+100;
        M=N;
        while(M)
        {
            t=M%10;

```

/******SPACE******/

```

        sum=sum+t*t*t;    /*本处可以写为 sum=sum+pow(t,3);*/
        M=M/10;
    }
    if(N==sum) find=1;
}
return N;
}

int max_value(int array[][4],int rw[])
{
    int i,j,max;
    max=array[0][0];
    for(i=0;i<3;i++)
        for(j=0;j<4;j++)
            if(max<array[i][j])
            {
                max=array[i][j];
                rw[0]=i;
                rw[1]=j;
            }
    return max;
}

void main()
{
    int array[3][4];        //用于存放水仙花数
    int row_col_num[2];     //用于存放最大水仙数的行列号
    int i,j;

    srand((unsigned)time(NULL));
    printf("\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<4;j++)
        {
            array[i][j]=narcissus();
            printf("%5d",array[i][j]);
        }
        printf("\n");
    }

    /*****SPACE*****/
    printf("Max=%d\n",max_value(array,row_col_num));
    printf("Col number of Max is:%5d\n",row_col_num[0]);
    printf("Row number of Max is:%5d\n",row_col_num[1]);
}

```

```
}
/*-----
```

【程序改错】

题目：主函数 main()中调用 fun()函数，其中 fun 函数的功能是将字符串 s 中下标为奇数的字符或 ASCII 码值为偶数的字符依次放入字符串 t 中。例如，字符串 s 为” AABBCCDDEEFF”，则输出应当是” ABBCDDEFF”。

下面给定的程序存在错误，请改正。

注意：不得增行或删行，也不得更改程序的结构。

```
-----*/
```

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#define N 80
void fun(char s[],char t[])
{
    unsigned i,j=0;
    for(i=0;i<strlen(s);i++)
/******FOUND*****/
        if(i%2||s[i]%2==0)
            t[j++]=s[i];
/******FOUND*****/
    t[j]='\0';
}

int main()
{
    char s[N],t[N];
    printf("Please input string s:");
/******FOUND*****/
    scanf("%s",s);
    fun(s,t);
    printf("The result is:%s\n",t);
    return 0;
}
```

```
/*-----
```

【程序设计】

题目：主函数 main() 中一维数组 num 为测试数据, 程序将测试数据转换为数字字符串并保存在数组 s 中, 然后将数字字符串 s 转换为货币格式保存到数组 t 中, 最终输出数组 t。

编写程序：

1. 编写函数

void LongtoStr(long m[], int n, char s[][30])

将数组 m 中 n 个整数转换成数字字符串, 保存在数组 s 中。

2. 编写函数

void Currency(char s[][30], int n, char M, char t[][30])

将 n 个数字字符串 s 转换为货币格式保存到数组 t 中并输出, M 表示货币符号。

例如：

测试数据：

12345678, 45678, 1689

运行结果：

num[0]=12345678 is changed as \$12,345,678

num[1]=45678 is changed as \$45,678

num[2]=1689 is changed as \$1,689

注意：请勿改动主函数 main() 中的任何语句。

-----*/

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<string.h>

#include<conio.h>

#define N 10

void LongtoStr(long m[], int n, char s[][30])

{

/******Program******/

int i, j, t, k;

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

{

 k=0;

 t=m[i];

 while(t)

 {

```

        s[i][k++]=t%10+'0';
        t=t/10;
    }
    s[i][k]=0;
    for(j=0;j<k/2;j++)
    {
        t=s[i][j];
        s[i][j]=s[i][k-1-j];
        s[i][k-1-j]=t;
    }
}
/*****      End      *****/

}

void Currency(char s[][30],int n,char M,char t[][30])
{
/*****Program*****/
int i,j,CD,tcd,sjcd;
for(i=0;i<n;i++)
{
    CD=strlen(s[i]);
    tcd=0;
    sjcd=0;
    while(sjcd!=CD)
    {
        t[i][0]=s[i][CD-1-sjcd];
        tcd++;
        sjcd++;
        for(j=tcd-1;j>=0;j--)
            t[i][j+1]=t[i][j];
        if(!(sjcd%3))
        {
            t[i][0]=';';
            tcd++;
            for(j=tcd-1;j>=0;j--)
                t[i][j+1]=t[i][j];
        }
    }
    t[i][0]=M;
    tcd++;
    t[i][tcd]=0;
}
/*****      End      *****/

```

```

}

int main()
{
    FILE *fp;
    long int num[N]={ 12345678,45678,1689};
    char s[N][30],t[N][30];
    int i,n;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("FILE open error.\n");
        exit(0);
    }

    n=3;
    LongtoStr(num,n,s);
    Currency(s,n,'$',t);

    for(i=0;i<n;i++)
    {
        printf("num[%d]=%ld is changed as %s\n",i,num[i],t[i]);
        fprintf(fp,"num[%d]=%ld is changed as %s\n",i,num[i],t[i]);
    }
    fclose(fp);

    return 0;
}
/*-----

```

【程序填空】

题目：以下程序将 5 个字符串从小到大进行排序。字符串比较的规则为：将两个字符串自左至右逐个字符按 ASCII 码值大小进行比较，直到出现不同的字符或遇到'\0'为止。如果全部字符相同，则认为相等；若出现不相同的字符，则以第一个不相同字符的比较结果为准。

注意：除要求填空的位置之外，请勿改动程序中的其他内容。
且不能使用 C 语言的库函数进行填空。

-----*/

```

#include <stdio.h>
#include <string.h>

```



```

#define N 5
#define MAXLEN 100

/*函数 compstr(char str1[],char str2[])实现字符串 str1 和
字符串 str2 的比较，若 str1 等于 str2,则返回 0；若 str1 小于
str2,则返回-1；若 str1 大于 str2,则返回 1*/

int compstr(char str1[],char str2[]){
    int flag=0;
    int i=0;
    while (str1[i]&&str2[i]&&!flag)
    {
        /*****SPACE*****/
        if (str1[i]>str2[i]) flag=1;
        else if (str1[i]<str2[i]) flag=-1;
        i++;
    }

    if(!str1[i] && str2[i]) flag=-1;
    else if(str1[i] && !str2[i]) flag=1;
    return flag;
}

void main(){
    char ls[N][MAXLEN]={"china","beijing","guangzhou","changchun","jiangsu"};
    char t[MAXLEN];
    int i,j;

    for(i=0;i<N-1;i++)
        for(j=0;j<N-i-1;j++)
            /*****SPACE*****/
            if(compstr(ls[j],ls[j+1])>0){
                strcpy(t,ls[j]);
                strcpy(ls[j],ls[j+1]);
                strcpy(ls[j+1],t);
            }
    printf("\n\n");
    for(i=0; i<N; i++) printf("%s ",ls[i]);
    printf("\n\n");
}
/*-----

```

【程序改错】

题目：主函数 main()中调用 fun()函数，其中 fun 函数的功能是：生成杨辉三角形的前六行。形如：

```
1
1  1
1  2  1
1  3  3  1
1  4  6  4  1
1  5 10 10 5  1
```

下面给定的程序存在错误，请改正。

注意：不得增行或删行，也不得更改程序的结构。

-----*/

```
#include <stdio.h>
#define N 6
/*****FOUND*****/
void yanghui(int a[][N])
{
    int i,j;
    for(i=0;i<N;i++)
    {
/*****FOUND*****/
        a[i][i]=1;a[i][0]=1;
    }
    for(i=2;i<N;i++)
/*****FOUND*****/
        for(j=1;j<=i-1;j++)
            a[i][j]=a[i-1][j-1]+a[i-1][j];
}
int main()
{
    int i,j,a[N][N];
    yanghui(a);
    for(i=0;i<N;i++)
    {
        for(j=0;j<=i;j++)
            printf("%4d",a[i][j]);
        printf("\n");
    }
}
```

/*-----

【程序设计】

题目：主函数 main() 中字符数组 s 为测试数据，程序将测试数据按以下规则复制到数组 t 中，再将数组 t 中的字符串逆置。规则如下：已知 n 为数组元素下标值，对于 n 为奇数位置的字符复制 n 次，对于 n 为偶数位置的字符只复制一次。

编写程序：

1. 编写函数 void CopyStr(char s[][30], int n, char t[][100]), 将 n 个字符串 s 按上述规则进行复制，结果保存在数组 t 中。
2. 编写 void Reserve(char t[][100], int n), 将数组 t 中 n 个字符串逆置。

测试数据：“ABCDEF”, “1234”

运行结果：

String ABCDEF is changed as FFFFEEDDDCBA

String 1234 is changed as 444321

注意：请勿改动主函数 main() 中的任何语句。

-----*/

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>
#define N 30
```

```
void CopyStr(char s[][30], int n, char t[][100])
```

```
{
```

```
/******Program*****/
```

```
int i, j, k, z, cd;
```

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

```
{
```

```
    k=0;
```

```
    cd=strlen(s[i]);
```

```
    for(j=0; j<cd; j++)
```

```
    {
```

```
        if(!(j%2))
```

```
            t[i][k++]=s[i][j];
```

```
    else
```

```
        {
```

```

                for(z=1;z<=j;z++)
                    t[i][k++]=s[i][j];
            }
        }
        t[i][k]=0;
    }
    /*****      End      *****/
}

```

```

void Reserve(char t[][100],int n)
{
    /*****Program*****/
    int i,j,k;
    char zc;
    for(i=0;i<n;i++)
    {
        k=strlen(t[i]);
        for(j=0;j<k/2;j++)
        {
            zc=t[i][j];
            t[i][j]=t[i][k-1-j];
            t[i][k-1-j]=zc;
        }
    }
    /*****      End      *****/
}

```

```

int main()
{
    char s[N][30]={ "ABCDEF","1234"};
    char t[N][100];
    int i,n=0;

    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("File open error\n");
        exit(0);
    }

    n=2;
    CopyStr(s,n,t);
    Reserve(t,n);
    for(i=0;i<n;i++)

```

```

    {
        printf("String %s is changed as %s\n",s[i],t[i]);
        fprintf(fp,"String %s is changed as %s\n",s[i],t[i]);
    }

    fclose(fp);

    return 0;
}
/*-----

```

【程序填空】

题目：程序实现在一个升序排列的整数数组中插入一个数，使数组仍保持升序排列。

注意：除要求填空的位置之外，请勿改动程序中的其他内容。
且不能使用 C 语言的库函数进行填空。

-----*/

```
#include <stdio.h>
```

```
/*输出数组元素*/
```

```
void print_array(int a[],int n) {
    int i;
    for(i=0;i<n;i++)
        printf("%4d", a[i]);
    printf("\n");
}
```

/*在数组中插入一个元素，使其仍然有序，x 为待插入的数，
n 为数组中原有元素的个数*/

```
void insert_array(int a[],int n,int x){
```

```
    int i=n-1;
    int find=0;
    while(i>=0 && !find){
        if(x<a[i])
        {
```

```
/******SPACE******/
```

```
            a[i+1]=a[i];
            i--;
        }
        else find=1;
```

```

    }
    a[i+1]=x;
}

void main(){
    int a[11]={1,3,5,8,11,13,17,21,26,30},x;

    printf("The array before insert:\n");
    print_array(a,10);
    printf("\nPlease Enter a Intger N=");
    scanf("%d",&x);
    insert_array(a,10,x);
    printf("The array after insert:\n");
    /*调用 print_array 输出插入元素后的数组*/
    /******SPACE******/
    print_array(a,11);
}
/*-----

```

【程序改错】

题目：主函数 main()调用 fun()函数，其中 fun 函数的功能是：计算二维数组周边元素之和，作为函数值返回。二维数组中的值在主函数中赋予。例如，若二维数组中的值为：

```

1  3  5  7  9
2  9  9  9  4
6  9  9  9  8
1  3  5  7  0

```

则函数值 $1+3+5+7+9+4+8+0+7+5+3+1+6+2=61$ 。

下面给定的程序存在错误，请改正。

注意：不得增行或删行，也不得更改程序的结构。

```

-----*/

#include<stdio.h>
/******FOUND******/
int fun(int a[][5])
{
    int i,j,sum=0;
    for(i=0;i<4;i++)
        for(j=0;j<4;j++)
/******FOUND******/

```

```

        if(i==0||j==0||i==3||j==4)
/*****FOUND*****/
        sum=sum+a[i][j];
    return sum;
}

int main()
{
    int aa[4][5]={ {1,3,5,7,9},{2,9,9,9,4},{6,9,9,9,8},{1,3,5,7,0}};
    int i,j,y;
    printf("The original data is:\n");
    for(i=0;i<4;i++)
    {
        for(j=0;j<5;j++)
            printf("%6d",aa[i][j]);
        printf("\n");
    }
    y=fun(aa);
    printf("The sum:%d\n",y);
    return 0;
}

```

/*-----

【程序设计】

题目：主函数 main()中一维数组 a 为测试数据，程序主要判断数组 a 中整数是否是完全数，并将完全数保存在数组 b 中，然后将数组 b 中最小的完全数移到第一个位置上,其他元素后移，其相对位置不变。

完全数：若一个正整数 x 的所有因子（包括 1，不包括 x 自身）之和等于 x 自身，则称 x 是一个完全数。例如：28 的因子有 1,2,4,7,14 且 $1+2+4+7+14=28$ ，因此 28 是一个完全数。

编写程序：

1. 编写函数 `int factor(int x[],int n,int y[])`，在数组 x 前 n 个数中查找完全数，将所有完全数保存在数组 y 中，函数返回完全数的个数。
2. 编写 `void MoveElement(int y[],int n)`，将数组 y 中 n 个元素最小的完全数移到第一个位置上，其他元素后移后相对位置不变。

测试数据: 5,496,23,28,6,8010,2012,1998,1975,2011

运行结果: Before Moving:

b[0]=496

b[1]=28

b[2]=6

After Moving:

b[0]=6

b[1]=496

b[2]=28

注意: 请勿改动主函数 main()中的任何语句。

-----*/

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<conio.h>
```

```
#include<ctype.h>
```

```
#include<string.h>
```

```
int factor(int x[],int n,int y[])
```

```
{
```

```
/******Program*****/
```

```
int i,j,k=0,s;
```

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

```
{
```

```
s=0;
```

```
for(j=1;j<x[i];j++)
```

```
if(!(x[i]%j))
```

```
s=s+j;
```

```
if(s==x[i])
```

```
y[k++]=x[i];
```

```
}
```

```
return k;
```

```
/****** End *****/
```

```
}
```

```
void MoveElement(int y[],int n)
```

```
{
```

```
/******Program*****/
```

```
int i,min=0,t;
```

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

```
if(y[i]<y[min])
```



```

        min=i;
while(min)
{
    t=y[0];
    for(i=0;i<n-1;i++)
        y[i]=y[i+1];
    y[i]=t;
    min--;
}
/*****      End      *****/
}

int main()
{
    int a[10]={5,496,23,28,6,8010,2012,1998,1975,2011},b[10]={0},n=10,m,i;

    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("File open error\n");
        exit(0);
    }

    m=factor(a,n,b);

    printf("Before Moving:\n"); //移动前
    fprintf(fp,"Before Moving:\n");
    for(i=0;i<m;i++)
    {
        printf("b[%d]=%d\n",i,b[i]);
        fprintf(fp,"b[%d]=%d\n",i,b[i]);
    }

    MoveElement(b,m);

    printf("After Moving:\n"); //移动后
    fprintf(fp,"After Moving:\n");
    for(i=0;i<m;i++)
    {
        printf("b[%d]=%d\n",i,b[i]);
        fprintf(fp,"b[%d]=%d\n",i,b[i]);
    }
    fclose(fp);
}

```

```
    return 0;
}
```

七、程序填空 共 1 题 （共计 20 分）

第 1 题 （20.0 分）

/*-----

【程序填空】

题目：使用冒泡法将 10 个随机生成的浮点数进行大到小的排序，
并将排好序的 10 个数分两行输出。

注意：除要求填空的位置之外，请勿改动程序中的其他内容。
且不能使用 C 语言的库函数进行填空。

-----*/

```
#include <stdio.h>
#define COUNT 10
#include <time.h>
#include <math.h>
#include <stdlib.h>

void bubble_sort(float f[],int n){
    int i,j;
    int swap=1;
    float t;
    i=0;
    while(i<n-1 && swap){
        swap=0;
        for(j=0;j<n-i-1;j++){
            /******SPACE*****/
            if(f[j]<f[j+1])
            {
                t=f[j];
                f[j]=f[j+1];
                f[j+1]=t;
                swap=1;
            }
        }
    }
}
```

```

        i++;
    }
}

void main(){
    int i;
    float f_data[COUNT];
    srand((unsigned)time(NULL));
    /*随机生成 10 个浮点数*/
    for(i=0;i<COUNT;i++)
        f_data[i]=(float)(rand()%900+100)/(float)10.0;
    printf("\n");

    /******SPACE*****/

    bubble_sort(f_data,COUNT);

    printf("The sorted 10 numbers;\n");

    for(i=0;i<COUNT;i++)
    {
        if(i%5==0) printf("\n");
        printf("%8.2f",f_data[i]);
    }
    printf("\n");
}

```

八、程序改错 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/*-----

【程序改错】

题目：主函数 main()调用 fun()函数，其中 fun 函数的功能是：找出 100 到 999 之间（含 100 和 999）所有整数中各位上数字之和为 x（x 为正整数）的整数并输出，符合条件的整数个数作为函数值返回。

例如，当 x 值为 5 时，满足条件的整数有：

104、113、122、131、140、203、212、221、230、302、311、320、401、410、500，一共 15 个。

下面给定的程序存在错误，请改正。

注意：不得增行或删行，也不得更改程序的结构。

-----*/

```
#include<stdio.h>
/*****FOUND*****/
int fun(int x)
{
    int n,s1,s2,s3,t;
    n=0;
    t=100;
    while(t<=999)
    {
/*****FOUND*****/
        s1=t%10; s2=t/10%10; s3=t/100;
        if(s1+s2+s3==x)
        {
            printf("%5d",t);
            n++;
        }
        t++;
    }
    return n;
}

int main()
{
    int x=-1;
    while(x<=0)
    {
        printf("Please input x(x>0):");
/*****FOUND*****/
        scanf("%d",&x);
    }
    printf("\nThe result is: %d\n",fun(x));
    return 0;
}
```

九、程序设计 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/*-----

【程序设计】

题目：主函数 main()中数字字符串 s 为测试数据，程序主要先对数字字符串 s 进行降序排序，将结果存入在数组 b 中，然后对数组 b 中数据进行压缩，不同数字保存在数组 b 中，数字出现次数保存在数组 times 中，计算各个数字加权和（即数字本身乘以该数字出现次数），结果保存在数组 c 中。

编写程序：

1. 编写函数 void SortAndChange(char s[],int n, int b[]),
对字符串 s 中 n 个字符进行降序排列,并将 n 个字符转换为数字，保存在数组 b 中。
2. 编写函数 int Calculcate(int b[],int n, int times[],int c[]),
对数组 b 中数字进行压缩，不同数字保存在数组 b 中，不同数字出现次数存入数组 times 中，并计算各个数字加权和，结果存入数组 c 中,函数返回不同数字的个数。

测试数据： 13334444555582298760

运行结果： b[0]=9 and times[0]=1, the weight sum is 9
 b[1]=8 and times[1]=2, the weight sum is 16
 ...
 b[9]=0 and times[9]=1, the weight sum is 0

注意：请勿改动主函数 main()中的任何语句。

-----*/

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>
#define N 50
```

```
void SortAndChange(char s[],int n,int b[])
{
/******Program*****/
int i,j,min,t;
for(i=0;i<n;i++)
{
    min=i;
```

```

        for(j=i+1;j<n;j++)
            if(s[j]>s[min])
                min=j;

        t=s[i];
        s[i]=s[min];
        s[min]=t;
    }
    for(i=0;i<n;i++)
        b[i]=s[i]-'0';
    /*****      End      *****/
}

```

```

int Calculcate(int b[],int n,int times[],int c[])
{
    /*****Program*****/
    int i,gs=0,cs=1;
    for(i=0;i<n;i++)
    {
        if(i!=(n-1)&&b[i+1]!=b[i])
        {
            times[gs]=cs;
            b[gs++]=b[i];
            cs=1;
        }
        else
            if(i!=(n-1)&&b[i+1]==b[i])
                cs++;
            else
                if(i==(n-1)&&b[i]!=b[i-1])
                {
                    times[gs]=cs;
                    b[gs++]=b[i];
                }
    }
    for(i=0;i<gs;i++)          /*计算各个数字加权和*/
        c[i]=times[i]*b[i];
    return gs;
    /*****      End      *****/
}

```

```

int main()
{
    char s[N]="13334444555582298760";
    int b[N]={0};
}

```

```

int c[10]={0};
int times[10]={0};
int num;
int i,n;

FILE *fp;
if((fp=fopen("DATA.TXT","w"))==NULL)
{
    printf("File open error\n");
    exit(0);
}

n=strlen(s);
SortAndChange(s,n,b);
num=Calculate(b,n,times,c);

for(i=0;i<num;i++)
{
    printf("b[%d] = %d and times[%d] =%d, the weight sum is %d\n",i,b[i],i,times[i],c[i]);
    fprintf(fp,"b[%d]=    %d    and    times[%d]    =%d,    the    weight    sum
is %d\n",i,b[i],i,times[i],c[i]);
}

fclose(fp);
getch();
return 0;
}

```

七、程序填空 共 1 题 （共计 20 分）

第 1 题 （20.0 分）

/*-----

【程序填空】

题目：将从键盘输入的 N 个整数储存在一个数组中,然后将数组中的数按从小到大的顺序进行排列。

注意：除要求填空的位置之外,请勿改动程序中的其他内容。
且不能使用 C 语言的库函数进行填空。

-----*/

```

#include "stdio.h"
#define N 5

/*以插入排序的方法将数组 a[]中的数据排序，当参数 order 为 0 时，
  将数据从小到大排列；当参数 order 为 1 时，将数据从大到小排列。*/
void insert_sort(int a[],int n,int order)
{
    int i,j,t;
    for(i=1;i<n;++i)
    {
        /*******SPACE*****/
        if(a[i]<a[i-1]&& order==0)      {
            t=a[i];
            j=0;
            for(j=i-1;j>=0 && t<a[j];--j)
                a[j+1]=a[j];
            a[j+1]=t;
        }

        if(a[i]>a[i-1] && order==1)
        {
            t=a[i];
            j=0;
            for(j=i-1;j>=0 && t<a[j];--j)
                a[j+1]=a[j];
            a[j+1]=t;
        }
    }
}

void main( ){
    int i,num[N];
    for( i=0;i<N;i++ )
    {
        printf("Enter the No. %3d Integer:\n",i);
        scanf("%d",&num[i]);
    }
    /*******SPACE*****/
    insert_sort(num,N,0);
    printf("\n\n");
    for( i=0;i<N;i++)
        printf("%5d",num[i]);
}

```



```
printf("\n\n");

}
```

八、程序改错 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/*-----

【程序改错】

题目：主函数 main()中调用 fun()函数，其中 fun 函数的功能是：
将字符串 s 正序和逆序进行连接，形成新的字符串 t。例如，
当字符串 s 为” ABCD” 时，形成的新串 t 为” ABCDDCBA”。

下面给定的程序存在错误，请改正。

注意：不得增行或删行，也不得更改程序的结构。

-----*/

```
#include<stdio.h>
#include<string.h>
/*****FOUND*****/
void fun(char s[],char t[])
{
    int i,d;
    d=strlen(s);
    for(i=0;i<d;i++) t[i]=s[i];
    for(i=0;i<d;i++) t[d+i]=s[d-1-i];
    /*****FOUND*****/
    t[2*d]='\0';
}

int main()
{
    char s[100],t[100];
    printf("Please enter string s:");
    /*****FOUND*****/
    scanf("%s",s);
    fun(s,t);
    printf("The result is:%s\n",t);
}
```

```
    return 0;
}
```

九、程序设计 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/*-----

【程序设计】

题目：主函数 main() 中二维数字字符串数组 s 为测试数据，程序将 n 个数字字符串转换为整数并输出，同时判断相邻两个整数是否是互满数对。

互满数对是指如果数 a 的真因子之和等于数 b，且数 b 的真因子之和等于数 a，则称(a,b)为一对互满数。例如，(220,284) 是一对互满数，因为 220 的真因子之和为 284（ $1+2+4+5+10+11+22+44+55+110=284$ ），且 284 的真因子之和为 220（ $1+2+4+71+142=220$ ）。

编写程序：

1. 编写函数 void chnum(char t[][20], int n, int b[]), 将 n 个数字字符串转换为整数，并保存到数组 b 中。
2. 编写函数 int allpairs(int b[], int n, int c[][2]), 判断数组 b 中 n 个元素相邻的两个整数是否是互满数，将互满数对保存到数组 c 中，函数返回互满数对的个数。

测试数据："2012","220","284","6232","6368"

运行结果：

(c[0][0],c[0][1])=(220,284)
(c[1][0],c[1][1])=(6232,6368)

注意：请勿改动主函数 main() 中的任何语句。

-----*/

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>
```

```

void chnum(char t[][20],int n,int b[])
{
/*****Program*****/
int i,s,j;
for(i=0;i<n;i++)
{
    s=0;
    j=0;
    while(t[i][j])
    {
        s=s*10+t[i][j]-'0';
        j++;
    }
    b[i]=s;
}
/***** End *****/
}

```

```

int allpairs(int b[],int n,int c[][2])
{
/*****Program*****/
int k=0,i,qyz,hyz,j;
for(i=1;i<n;i++)
{
    qyz=hyz=0;
    for(j=1;j<b[i-1];j++)
        if(b[i-1]%j==0)
            qyz=qyz+j;
    for(j=1;j<b[i];j++)
        if(b[i]%j==0)
            hyz=hyz+j;
    if((qyz==b[i])&&(hyz==b[i-1]))
    {
        c[k][0]=b[i-1];
        c[k++][1]=b[i];
    }
}
return k;
/***** End *****/
}

```

```

int main()

```

```

{
    char s[][20]={ "2012","220","284","6232","6368"};
    int b[5]={0},n=5,i,num;
    int c[50][2];
    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("File open error\n");
        exit(0);
    }
    chnum(s,n,b);
    num=allpairs(b,n,c);
    for(i=0;i<num;i++)
    {
        printf("(c[%d][0],c[%d][1])=(%d,%d)\n",i,i,c[i][0],c[i][1]);
        fprintf(fp,"(c[%d][0],c[%d][1])=(%d,%d)\n",i,i,c[i][0],c[i][1]);
    }
    fclose(fp);
    getch();
    return 0;
}

```

七、程序填空 共 1 题 （共计 20 分）

第 1 题 （20.0 分）

/*-----

【程序填空】

-----*/

题目：统计从键盘输入的一行字符中单词的个数，单词之间用空格分开。

注意：除要求填空的位置之外,请勿改动程序中的其他内容。
且不能使用 C 语言的库函数进行填空。

-----*/

```
#include <stdio.h>
```

```
int word_stat(char string[])
```

```
{
    char c;
```

```

    int i,word=0;
    int wnum=0;
    /*****SPACE*****/
    for (i=0;c=string[i];i++)
        if(c==' ') word=0;
        else if(word==0)
        {
            word=1;
            wnum++;
        }
    /*****SPACE*****/
    return wnum;
}

void main()
{
    char string[100];
    printf("Please enter a string:\n");
    gets(string);
    printf("The number of words is: %d\n",word_stat(string));
}

```

八、程序改错 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/*-----

【程序改错】

题目：主函数 main()调用 fun()函数，其中 fun 函数的功能是：求表达式

$s = aa \dots aa - \dots - aaa - aa - a$

（此处 $aa \dots aa$ 表示 n 个 a ， a 和 n 的值为 1~9）

例如： $a=3$ ， $n=6$ ，则以上表达式为：

$s=333333-333333-33333-333-33-3$

其值是 296298。其中 a 和 n 是 fun 函数的形参，表达式的值作为函数值返回 main 函数。

下面给定的程序存在错误，请改正。

注意：不得增行或删行，也不得更改程序的结构。

-----*/

```

#include<stdio.h>
long fun(int a,int n)
{
    int j;
    /*******FOUND*****/
    long s=0,t=0;
    for(j=0;j<n;j++)
    /*******FOUND*****/
        t=t*10+a;
    s=t;
    for(j=1;j<n;j++)
    {
    /*******FOUND*****/
        t=t/10;
        s=s-t;
    }
    return s;
}

int main()
{
    int a,n;
    printf("Please enter a and n:");
    scanf("%d%d",&a,&n);
    printf("The value of function is %ld\n",fun(a,n));
    return 0;
}

```

九、程序设计 共 1 题 （共计 30 分）

/*-----
【程序设计】

题目：主函数 main()主要对仅由数字与字母组成的字符串中字符按下列规则重新排列：字母从左向右排列在字符串左边，数字从字符串的右边开始向左排列，并判断该字符串中数字字符串是否是回文字符串。若是回文字符串，则输出该回文数字字符串及其对应的原始字符串。

回文字符串是指一个字符串和该字符串的逆序相同的字符串。

例如：

若字符串为"glbc3d3cbg1"，重新排列后的字符串为"gbcdcbg1331"，

"1331"为回文字符串，输出。

若字符串为"ab156ba"，重新排列后的字符串为"abba651"，

"651"不是回文字符串，不输出。

编写程序：

1. 编写函数

```
void adjust(char s[][20],int n,char t[][20]),
```

将 n 个数字字母组合的字符串按上述规则进行排列，并
保存到数组 t 中，不得改变原始字符串 s 内容。

2. 编写函数

```
int JudgeHuiwen(char t[][20],int n,int loc[],char huiwen[][20]),
```

主要判断数组 t 中 n 个字符串所包含的数字字符串是否
是回文字符串，将回文数字字符串保存到数组 huiwen
中，其对应的原始字符串在数组中的位置保存到数组
loc 中。

测试数据："glbc3d3cbg2","abc4abc","glbc3d3cbg1","ab156ba","ab122ba1"

运行结果：

The number string 4 in abc4abc is huiwen string!

The number string 1331 in glbc3d3cbg1 is huiwen string!

The number string 1221 in ab122ba1 is huiwen string!

注意：请勿改动主函数 main()中的任何语句。

-----*/

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<conio.h>
```

```
#include<ctype.h>
```

```
#include<string.h>
```

```
#define N 10
```

```
void adjust(char s[][20],int n,char t[][20])
```

```
{
```

```
/******Program******/
```

```
int i,j,k;
```

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

```
{
```

```
    k=0;
```

```

        for(j=0;s[i][j];j++)
            if(isalpha(s[i][j]))
                t[i][k++]=s[i][j];
        for(;j>=0;j--)
            if(isdigit(s[i][j]))
                t[i][k++]=s[i][j];
        t[i][k]=0;
    }
    /*****      End      *****/
}

int JudgeHuiwen(char t[][20],int n,int loc[],char huiwen[][20])
{
    /*****Program*****/
    /*主要判断数组 t 中 n 个字符串所包含的数字字符串是否
        是回文字符串，将回文数字字符串保存到数组 huiwen
        中，其对应的原始字符串在数组中的位置保存到数组
        loc 中。*/

    int flag;
    int i,j,k=0,s,w,m,hw;
    for(i=0;i<n;i++)
    {
        j=0;
        while(t[i][j])
        {
            if(isdigit(t[i][j]))
            {
                s=j;
                w=j;
                while(isdigit(t[i][w]))
                    w++;
                w=w-1;
                flag=1;
                while(s<w)
                {
                    if(t[i][s]!=t[i][w])
                    {
                        flag=0;
                        break;
                    }
                    s++;
                    w--;
                }
                if(flag)

```



```

        {
            loc[k]=i;
            for(m=0,hw=0;t[i][m];m++)
                if(isdigit(t[i][m]))
                    huiwen[k][hw++]=t[i][m];
            huiwen[k][hw]=0;
            k++;
        }
        break;
    }
    j++;
}

}
return k;
/*****      End      *****/
}

int main()
{
    char s[N][20]={ "g1bc3d3cbg2","abc4abc","g1bc3d3cbg1","ab156ba","ab122ba1"};
    char t[N][20],huiwen[N][20];
    int loc[N]={0};
    int num=0,n,i;

    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("File open error\n");
        exit(0);
    }

    n=5;
    adjust(s,n,t);
    num=JudgeHuiwen(t,n,loc,huiwen);
    for(i=0;i<num;i++)
    {
        printf("The number string %s in %s is huiwen string!\n",huiwen[i],s[loc[i]]);
        fprintf(fp,"The number string %s in %s is huiwen string!\n",huiwen[i],s[loc[i]]);
    }
    fclose(fp);
    getch();
    return 0;
}

```

七、程序填空 共 1 题 （共计 20 分）

第 1 题 （20.0 分）

/*-----

【程序填空】

功能:十个小孩围成一圈分糖果，老师分给第一个小孩 10 块，第二个小孩 2 块，第三个小孩 8 块，第四个小孩 22 块，第五个小孩 16 块，第六个小孩 4 块，第七个小孩 10 块，第八个小孩 6 块，第九个小孩 14 块，第十个小孩 20 块。然后所有的小孩同时将自己手中的糖分一半给左边的小孩；糖块数为奇数的人可向老师要一块。问经过这样几次调整后大家手中的糖的块数都一样？每人各有多少块糖？

-----*/

```
main()
{
    int i,count=0,a[11]={0,10,2,8,22,16,4,10,6,14,20} ;
    while(1)
    {
        for(i=1 ; i<=10 ; i++)
            a[i-1]=a[i-1]/2+a[i]/2 ;
        /*****SPACE*****/
        a[0]=0;
        for(i=1 ; i<=10 ; i++)
            /*****SPACE*****/
            if(a[i]%2) a[i]++ ;
        for(i=1 ; i<10 ; i++)
            /*****SPACE*****/
            if(a[i]!=a[i+1]) break;
        if(i==10)
            break ;
        else
        {
            a[0]=0 ;
            count++ ;
        }
    }
    printf("count=%d number=%d\n",count,a[1]) ;
}
```

八、程序改错 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/*-----

【程序改错】

下列程序的功能是：将一个形如字符 1-字符 2 的字符串进行扩展，
例如"a-d"字符串扩展成"abcd","4-9"字符串扩展在"456789"。

其中字符 1、字符 2 必须形式统一（同为大写字母、同为小写字母、
同为数字字符，且满足字符 1<字符 2，否则不予扩展）。

例如，字符串"asg-l3-7\$WX-ZA-9=2-7r-w"进行扩展后
为"asghijkl34567\$WZYZA-9=234567rstuvw"。

下列给定的程序存在错误，请改正。

-----*/

```
#include "stdio.h"
#include<string.h>
int judge(char p,char n)
{
    if(p>='0'&& p<n &&n<='9') return 1;
    if(p>='a'&& p<n &&n <='z') return 1;
    if(p>='A'&& p<n &&n <='Z') return 1;
    return 0;
}
void expend(char str[],int pos,int len)
{
    char t[200],c;
    int i,j;
    for(i=0;i<pos;i++)
        t[i]=str[i];
    /******FOUND*****/
    c=str[pos-1];
    for(;i<pos+len;i++)
        t[i]=++c;
    for(j=pos+1;str[j]!='0';i++,j++)
        t[i]=str[j];
    t[i]=0;
    /******FOUND*****/
    strcpy(str,t);
}
```

```
void main()
{
    char str[200]="asg-l3-7$WX-ZA-9=2-7r-w";
    int i=1,n;
```

```

while(str[i]!=0)
{
    /*****FOUND*****/
    if(str[i]=='-'&&judge(str[i-1],str[i+1]))
    {
        n=str[i+1]-str[i-1]-1;
        expend(str,i,n);
        i=i+n-1;
    }
    i++;
}
printf("%s\n",str);
}

```

九、程序设计 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/*-----

【程序设计】

程序功能：在去除字符串非字母字符，并忽略字母大小写的情况下，判断所给字符串是否为回文。程序中允许使用系统提供的字符串处理函数。
编写程序：

1. 编写 void movechar(char str[])函数，去除 str 中的非字母字符。
2. 编写 void ishw(char str[][100],int n) 函数，判断 str 中 n 个字符串是否为回文，如是则在字符串前部加上字符串"is:"，否则加上"no:"。

-----*/

答案：存在 data.txt 文件，且文件内容为：

is:SxPfygaagyFPxS

no:RdafGKInKcekeKnlKGfadR

no:xbifXyHVRqqTLLTqRVHyXfibx

is:evEeRzzReEve

no:vbeThisUUVqqRTTRppVUUsihTebv

no:hxrskGjLKGgkLjGkrOxh

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
void movechar(char str[])
{

```

```

/*****Program*****/
int i,k=0;
for(i=0;str[i];i++)
    if(isalpha(str[i]))
        str[k++]=str[i];
str[k]=0;
/***** End *****/
}

void ishw(char str[][100],int n)
{
/*****Program*****/
int i,j,k,flag;
for(i=0;i<n;i++)
{
    flag=1;
    movechar(str[i]);
    k=strlen(str[i])-1;
    for(j=0;j<k;j++,k--)
        if(tolower(str[i][j])!=tolower(str[i][k]))
            flag=0;
    for(j=strlen(str[i]);j>=0;j--)
        str[i][j+3]=str[i][j];
    str[i][2]='.';
    if(flag)
    {
        str[i][0]='i';
        str[i][1]='s';
    }
    else
    {
        str[i][0]='n';
        str[i][1]='o';
    }
}
/***** End *****/
}

void main()
{
    char str[6][100]={ {"Sx,Pfyga:agyFPxS!"},
                        {"Rdaf GKI n! Kcek'e:KnlKGfadR"},
                        {"xbifXyHVRqq10TLL98TqRVHyXfibx"},
                        {"10evEeRz454zReEve01"},
                        {"vbe,This UUVqqRTTRppVUU,sihTebv"},
                        {"hx()r;skGj: LKGGkL!jGkrOxh"}
    };

```

```

int i;
FILE *fp;
ishw(str,6);
if((fp=fopen("DATA.TXT","w"))==NULL)
{ printf("File open error\n");
  exit(0);
}
for(i=0;i<6;i++)
{   printf("%s\n",str[i]);
    fprintf(fp,"%s\n",str[i]);
}
fclose(fp);
}

```

七、程序填空 共 1 题 （共计 20 分）

第 1 题 （20.0 分）

/*-----*/

【程序填空】

程序功能：输入两个字符串 searched 和 search，判断字符串 search 是否是字符串 searched 的子串，如果是，则输出 search 在 searched 中起始位置，如果不是，则输出-1（所谓 search 是 searched 的子串，即 search 是 searched 的一部分）。本题假定字符串 search 的长度小于等于字符串 searched 的长度。

源程序如下：

```

-----*/
#include <stdio.h>
#include <string.h>
int find(char sestr[],char sedstr[])
{
    int selen;
    int sedlen;
    int i,j;
    selen=strlen(sestr);
    sedlen=strlen(sedstr);
    for(i=0;i<=sedlen-selen;i++)
    {
        j=0;
        while(sestr[j]==sedstr[i+j]&&sestr[j]!=0)
            j++;
    }
}

```

```

        if(senstr[j]==0)
            return i;
    }
    return -1;
}

int main()
{
    char searched[100], search[20];
    int f;
    gets(searched);
    gets(search);
    f=find(search,searched);
    if(f==-1)printf("no substring\n");
    else printf("is substring\n");
    return 0;
}

```

八、程序改错 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/*-----

【程序改错】

程序功能：程序将一个字符串后接一个整数，形成一个新的字符串。其中函数 **change** 将数字 **n** 转换为数字字符串存放在数组 **s** 中；函数 **link** 将字符串 **t** 连接到字符串 **s** 后。

下列给定的程序存在错误，请改正。

-----*/

```

#include<stdio.h>
#include<stdlib.h>
void change(long n,char s[])
{
    long i,r=0;
    while(n!=0)
    {
        r=r*10+n%10;

```

/******FOUND******/

```

        n=n/10;
    }
    i=0;
    while(r!=0)
    {
        s[i++]=r%10+'0';
        r=r/10;
    }
    /*******FOUND*****/
    s[i]=0;
}
void link(char s[],char t[])
{
    long i=0,j;
    while(s[i]!=0)
        i++;
    j=0;
    /*******FOUND*****/
    for(;s[i++]=t[j++];);
}
int main()
{
    char s[100]={"break"},t[100];
    long num=12345;
    change(num,t);
    link(s,t);
    printf("%s\n",s);
    return 0;
}

```

九、程序设计 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/*-----

【程序设计】

程序对形如"12%u&770_<j80! @--2352fjld"的字符串进行处理，提取其中的数字字符转换为一个整数，重复出现的数字字符只提取一次。

编写程序：

1、 编写 long abstract(char str[])函数，处理形如

"12%u&770_<j80! @--2352fjld"的字符串，将其中的数字字符提取出来，形成一个整数返回。但重复出现的数字字符只取一个，即从上述字符中提取的整数位 1270835，并将该整数返回。

2、编写 int single(int n)函数，判断 n 是否是首次出现在字符串中的数字，如是，返回真，否则返回假。

----- */

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
long abstract(char str[])
{
    /*****Program*****/
    int sz[10]={0},i;
    long s=0;
    for(i=0;str[i];i++)
        if(isdigit(str[i]))
        {
            if(sz[str[i]-'0']>0)
                continue;
            else
            {
                s=s*10+str[i]-'0';
                sz[str[i]-'0']++;
            }
        }
    return s;
    /***** End *****/
}

int single(int n,char str[])
{
    /*****Program*****/
    int i;
    for(i=0;str[i];i++)
        if(isdigit(str[i]))
            if((str[i]-'0')==n)
                return 1;
            else
                return 0;
    /***** End *****/
}

int main()
{
```

```

char str[100]={ "12%u&770_<j80! @--2352fjld"};
long n;
FILE *fp;
n=abstract(str);
if((fp=fopen("data.txt","w"))==NULL)
{printf("File open error\n");
exit(0);
}
printf("%10ld\n",n);
fprintf(fp,"%10ld\n",n);
fclose(fp);
return 0;
}

```

七、程序填空 共 1 题 （共计 20 分）

第 1 题 （20.0 分）

/*-----

【程序填空】

程序功能：在一行英文文本中查找给定的单词。
 文本由若干个单词组成，单词之间由空格间隔。
 函数 find 完成在 s 字符串中查找 t 中包含的单词。
 如找到，函数返回 t 字符串出现的次数，并将所有出现的位置保存在数组 pos 中。
 源程序如下：

-----*/

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>
int find(char s[],char t[],int p[])
{
    int i,j,k=0;
    char w[20];
    for(i=0;s[i]!='\0';i++)
    {    if(isalpha(s[i]))
    {
        for(j=0;s[i+j]!=' ';&&s[i+j]!='\0';j++)
            w[j]=s[i+j];
        w[j]=0;

```

```

        if(strcmp(w,t)==0)    p[k++]=i;
        i=i+j;
    }
}
return k;
}
int main()
{
    char text[100]="Xian an ancient city in landlocked Shaanxi Province",word[20];
    int i,j,pos[20];
    printf("输入要查找的单词");
    gets(word);
    i=find(text,word,pos);
    if(i!=0)
        for(j=0;j<i;j++)
            printf("text[%d]\n",pos[j]);
    return 0;
}

```

八、程序改错 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/*-----

【程序改错】

程序功能：下面左侧的一个 4*4 的数组元素
经程序处理后成为右侧的数组形式。

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

1	5	3	13
2	6	10	8
9	7	11	15
4	14	12	16

下面程序存在错误，请改正。

-----*/

```

#include "stdio.h"

```

```

int main()
{
    int a[4][4]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16},m,k,t;
    for(m=0;m<4;m++)
    /******FOUND******/
        for(k=1;k<4;k++)
    /******FOUND******/
        if((m+k)%2)
            t=a[m][k];a[m][k]=a[k][m];a[k][m]=t;
    for(m=0;m<4;m++)
    {   for(k=0;k<4;k++)
    /******FOUND******/
        printf("%3d",a[m][k]);
        printf("\n");
    }
    return 0;
}

```

九、程序设计 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/*-----

【程序设计】

程序功能：二维字符数组 `str` 中的若干个字符串长度不等，以最长的字符串长度为准，小于该长度的字符串前部添加若干个“*”，使得所有字符串都等长，程序中不得使用系统提供的任何字符处理函数。

编写程序：

1. 编写 `int getmaxlen(char str[][20],int n,int len[])`，函数，求出 `str` 中每个字符串的实际长度值并存入数组 `len` 中，将字符串的最大长度返回。
2. 编写 `void fillstar (char str[][20], int len[], int n, int maxlen)` 函数，将 `str` 数组中长度小于 `maxlen` 的字符串前部填充若干字符“*”，填充后使得字符串长度等于 `maxlen`，`str` 数组中各字符串的长度保存在 `len` 数组对应的元素中。

主函数如下：

-----*/

```

#include "stdio.h"

```

```

#include "stdlib.h"
int  getmaxlen(char str[][100],int n,int len[])
{
    /*******Program*****/
    int i,k;
    for(i=0;i<n;i++)
    {
        k=0;
        while(str[i][k++] );
        len[i]=k-1;
    }
    k=0;
    for(i=1;i<n;i++)
        if(len[i]>len[k])
            k=i;
    return len[k];
    /******* End *****/
}
void fillstar(char str[][100],int len[],int n,int maxlen)
{
    /*******Program*****/
    int i,j,k;
    for(i=0;i<n;i++)
    {
        k=maxlen-len[i];
        for(j=len[i];j>=0;j--)
            str[i][j+k]=str[i][j];
        for(j=0;j<k;j++)
            str[i][j]='*';
    }
    /******* End *****/
}
int main()
{
    char str[5][100]={ {"error"}, {"getmaxlen"}, {"document"}, {"missing"}, {"and"} };
    int len[5],i,max;
    FILE *fp;
    max=getmaxlen(str,5,len);
    fillstar(str,len,5,max);
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("file open error\n");
        exit(0);
    }
}

```

```

        for(i=0;i<5;i++)
        {   printf("%s\n",str[i]);
            fprintf(fp,"%s\n",str[i]);
        }
        fclose(fp);
        return 0;
    }
/*-----

```

【程序填空】

题目：输入一字符串，检查字符串是否满足以下条件：

- (1) 字符串 s 中左括号 “(” 的个数与右括号 “)” 的个数相同；
- (2) 从字符串 s 的首字符起顺序查找右括号 “)” 的个数在什么时候均不超过所遇到的左括号 “(” 的个数。若字符串同时满足上述两个条件，函数返回 1，否则返回 0。

注意：请勿改动主函数 main() 中的其他内容。

```

-----*/

#include<stdio.h>
#include<string.h>
void main()
{
    int fun(char s[]);
    char s[80];
    int c;
    printf("Please input the string:");
    gets(s);
    c=fun(s);
    printf("c=%d\n",c);
}
int fun(char s[])
{
    int c1,c2,i;
    i=c1=c2=0;
    while(s[i]!='\0')
    {
        if(s[i]=='(') c1++;
        else if(s[i]==')')

```

```

        {
            c2++;
/*****SPACE*****/
            if(c1<c2) return(0);
        }
        i++;
    }
/*****SPACE*****/
    return(c1==c2);
}

/*-----
【程序改错】
-----

```

题目：下面程序中，函数 fun 的功能是：在任意给定的 9 个正整数中找出按升序排列时处于中间的数，将原数据序列中比该中间数小的数用该中间数替换，位置不变，在主函数中输出处理后的数据序列，并将中间数作为函数值返回。

例如，有 9 个正整数：1 5 7 23 87 5 8 21 45
 按升序排列时的中间数为：8
 处理后主函数中的输出为：8 8 8 23 87 8 8 21 45

下面给定的程序存在错误，请改正。

```

-----
注意：不得增行或删行，也不得更改程序的结构。
-----*/

```

```

#include <stdio.h>
#define N 9
int find(int []);
void sort(int []);
void main()
{
    int i, x[N]={ 1,5,7,23,87,5,8,21,45};
    for(i=0; i<9;i++)
        printf("%5d",x[i]);
    printf("\n");
/*****FOUND*****/
    printf("The mid number is:%5d\n",find(x));
    for(i=0; i<9;i++)
        printf("%5d",x[i]);
    printf("\n");
}

```

```

}
int find(int x[])
{
    int i,mid,b[N];
    for(i=0;i<N;i++)
        b[i]=x[i];
    sort(b);
    mid=b[N/2];
    for(i=0; i<N;i++)
        /******FOUND*****/
        if(x[i] < mid) x[i]=mid;
    return  mid;
}

void sort(int b[])
{
    int i,j,k,t;
    for(i=0;i<N;i++)
    {
        k=i;
        for(j=i+1;j<N;j++)
            /******FOUND*****/
            if(b[k]>b[j]) k=j;
        if(k != i )
        {
            t=b[i]; b[i]=b[k]; b[k]=t;
        }
    }
}
/*-----

```

【程序设计】

题目：主函数 main()中由数字与字母组成的字符串 s 为测试数据，程序主要将字符串 s 中字符复制到字符数组 t 中，下标为偶数位置上的字母字符复制两次，其余字符复制一次；然后判断字符串 t 中所包含数字子串对应的整数是否是"平方镜反数"。

例如：字符串 s 为"gl2bc32fA31"，按上述规则，则字符串 t 为"gg12bcc32fAA31"。

平方镜反数：

该数的平方与该数的反序数的平方互为反序数。

例如：12 的平方等于 144，12 的反序数 21 的平方等于 441，则 12 就是一个"平方镜反数"。

编写程序：

1. 编写函数 void GenStr(char s[],char t[]),按上述要求生成新的字符串，结果保存在字符数组 t 中。
2. 编写函数 int TestFanshu(char t[],int b[]),判断字符串 t 中所包含数字子串对应的整数是否是平方镜反数，将平方镜反数保存在数组 b 中，函数返回平方镜反数的个数。

注意：请勿改动主函数 main()中的任何语句。

-----*/

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>
#define N 80

void GenStr(char s[],char t[])
{
    /******Program*****/
    int i,j,k=0;
    for(i=0;s[i];i++)
    {
        if(!(i%2)&&((s[i]>='a'&&s[i]<='z')||(s[i]>='A'&&s[i]<='Z'))))
            for(j=0;j<2;j++)
                t[k++]=s[i];
        else
            t[k++]=s[i];
    }
    t[k]=0;
    /****** End *****/
}

int TestFanshu(char t[],int b[])
{
    /******Program*****/
    int k=0,sz,bc,nx,i;
    for(i=0;t[i];i++)
```

```

{
    if(!(t[i]>='0'&&t[i]<='9'))
    {
        sz=0;
        nx=0;
    }
    if(t[i]>='0'&&t[i]<='9')
    {
        sz=sz*10+t[i]-'0';
    }
    if((t[i]>='0'&&t[i]<='9')&&(t[i+1]<'0'||t[i+1]>'9'))
    {
        bc=sz;
        while(bc)
        {
            nx=nx*10+bc%10;
            bc=bc/10;
        }
        nx=nx*nx;
        bc=0;
        while(nx)
        {
            bc=bc*10+nx%10;
            nx=nx/10;
        }
        if(sz*sz==bc)
        {
            b[k++]=sz;
        }
    }
}

return k;
/*****      End      *****/
}

int main()
{
    char s[]="g12bc32fA31";
    char t[N];
    int b[N]={0};
    int num=0,i=0;

    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)

```

```

    {
        printf("File open error\n");
        exit(0);
    }

    GenStr(s,t);
    printf("The string t is: %s\n",t);
    fprintf(fp,"The string t is: %s\n",t);

    num=TestFanshu(t,b);
    printf("The pingfangjingfanshu is: ");
    fprintf(fp,"The pingfangjingfanshu is: ");

    for(i=0;i<num;i++)
    {
        printf("%d ",b[i]);
        fprintf(fp,"%d ",b[i]);
    }
    printf("\n");
    fprintf(fp,"\n");

    fclose(fp);

    return 0;
}
/*-----

```

【程序填空】

题目：对二维数组 **a** 中存储的 **N*N** 矩阵做如下操作：先将每一行中值最大的元素与该行位于主对角线处的元素交换，然后对主对角线上所有元素排序使其自左上角到右下角升序排列，最后输出排序后主对角线上各元素的值。

注意：请勿改动主函数 **main()** 中的其他内容。

-----*/

```

#include<stdio.h>
#define N 5
void fun(int x[][N],int n)
{
    int i,j,t,arr,col;
    for(i=0;i<n;i++)

```

```

    {
        arr=x[i][0];col=0;
        for(i=0;i<n;i++)
        {
            arr=x[i][0];col=0;
/*****SPACE*****/
            for(j=0;j<n;j++) if(x[i][j]>=arr) {arr=x[i][j]; col=j;}
            t=x[i][i]; x[i][i]=x[i][col];x[i][col]=t;
        }
        for(i=0;i<n-1;i++)
            for(j=i+1;j<n;j++)
/*****SPACE*****/
                if(x[i][i]>x[j][j])
                    { t=x[i][i]; x[i][i]=x[j][j]; x[j][j]=t;}
    }
}
void main()
{
    int a[N][N]={ { 10,25,24,13,23},{ 11,22,12,21,14},{ 20,15,19,16,18},
                  { 17,9,4,5,3},{ 2,1,6,7,8}};

    int i;
    fun(a,N);
    for(i=0;i<N;i++) printf("%4d",a[i][i]);
    printf("\n");
}

```

/*-----

【程序改错】

题目：下面的程序中，函数 fun 的功能是:从低位开始取出长整型变量 s 中奇数位上的数，依次构成一个新数放在 t 中。高位仍在高位,低位仍在低位。例如，当 s 中的数为：654321 时，t 中的数为：531。

请改正程序中的错误，使它能得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构。

-----*/

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

```

```

#include <math.h>

/*****FOUND*****/
long fun (long s);
long getnumber();
void main()
{
    long s,t;
    srand((unsigned)time(NULL));    //初始化随机数时间序列
    s=getnumber();                  //随机获取一个长整数
    t=fun(s);
    printf("The result is: %ld,%ld\n", s,t);
}
long fun (long s)
{
    long t=0;
/*****FOUND*****/
    long step=pow(10,t);
    while (s > 0)
    {
        t=t+s%10*step;
        step=10*step;
/*****FOUND*****/
        s = s/100;
    }
    return t;
}
long getnumber()
{
    int i,k;
    long n=0;
    k=rand()%4+5;
    for(i=0;i<k;i++)
    {
        while(n==0)
            n=10*n+rand()%10;
        n=10*n+rand()%10;
    }
    return n;
}
/*****

```

题目：主函数 main()中由数字和字母组成的字符串 s 为测试数据，程序主要提取字符串 s 中由字符'\$'分隔的不同八进制数形式的字符串，

将八进制数形式字符串所对应的字符生成新的字符串 t; 然后统计字符串 t 中所有数字子串对应整数的出现频率。

例如: 字符串 s 为"101\$062\$063\$114\$066\$110\$062\$063\$141\$071", 处理后字符串 t 为"A23L6H23a9", 其中"23"对应整数 23 出现频率为 2, "6"和"9"对应整数出现频率为 1。

编写程序:

1. 编写函数 void GenStr(char s[],char t[]),计算字符串 s 中不同八进制数形式的字符串, 将每个八进制数形式字符串所对应的字符保存在字符数组 t 中。
2. 编写函数 int CountFreq(char t[],int b[],int Freq[]), 统计字符数组 t 中不同数字子串对应整数出现频率, 不同整数保存在整数数组 b 中,出现频率保存在整数数组 Freq 中, 函数返回整数个数。

注意: 请勿改动主程序 main()中的任何语句。

*****/

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>
#define N 10
```

```
void GenStr(char s[],char t[])
{
    /*******Program*****/
    int i,k=0,n=0;
    for(i=0;s[i];i++)
    {
        if(isdigit(s[i]))
            k=k*8+s[i]-'0';
        if(isdigit(s[i])&&(!isdigit(s[i+1])))
        {
            t[n++]=k;
            k=0;
        }
    }
    t[n]=0;
    /******* End *****/
}
```

```

}

int CountFreq(char t[],int b[],int Freq[])
{
    /*******Program*****/
    /*编写函数 int CountFreq(char t[],int b[],int Freq[]), 统计
    字符串 t 中不同数字子串对应整数出现频率, 不同整数保存在
    整数数组 b 中,出现频率保存在整数数组 Freq 中, 函数返回整数个
    数。*/
    int i,j,z,k=0,num,flag;
    char bc[20];
    for(i=0;t[i];i++)
    {
        num=0;flag=1;;
        if(isdigit(t[i])) /*当运行到数字字符的时候, 进行如下处理*/
        {
            for(j=0;isdigit(t[i]);i++,j++) /*无影响对即将连续的字符串进行保存处理*/
            {
                num=num*10+t[i]-'0'; /*将连续的字符串保存为数据为下面存入数
组 Freq 做准备*/
                bc[j]=t[i]; /*将连续的字符串保存为下面查找到相同的字
符串来计数做准备*/
            }
            bc[j]=0;
            for(j=0;j<k;j++) /*到 Freq 中查看当前取出的字符串所对应的
数据是否已经出现: 控制其为不重复元素*/
            if(num==Freq[j])
            {
                flag=0;
                break;
            } /*如果 flag 为 0, 则说明当前取出的数据
已经出现*/
            if(flag)
            {
                Freq[k]=num; /*如果这是一个新的字符, 那么就在此保存
*/
                b[k]=0; /*同时将此位置的次数置 0*/
                for(j=0;j<strlen(t)-strlen(bc)+1;j++) /*从最前面开始查找与之相同的字符串*/
                {
                    for(z=0;bc[z];z++) /*断开此处依次比较如果不同就退出以便下
面对是否为已经存储的字符串做准备*/
                    if(bc[z]!=t[j+z])
                        break;
                    if(!bc[z]) /*对刚刚走过的部分进行是否为已经保存的

```

字符串的判断*/

```
        {
            b[k]++;
            j=j+z;
        }
    }
    k++;
    /*存储完毕为下一次做准备*/
}

}
}
return k;
/***** End *****/
}
```

```
int main()
{
    char s[]="101$062$063$114$066$110$062$063$141$071";
    char t[N]="";
    int b[N]={0},Freq[N]={0};
    int num=0,i=0;

    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("File open error\n");
        exit(0);
    }

    GenStr(s,t);
    printf("The string t is: %s\n",t);
    fprintf(fp,"The string t is: %s\n",t);

    num=CountFreq(t,b,Freq);
    printf("The different integer and frequency is: ");
    fprintf(fp,"The different integer and frequency is: ");

    for(i=0;i<num;i++)
    {
        printf("[%d: %d] ",b[i],Freq[i]);
        fprintf(fp,"%d: %d] ",b[i],Freq[i]);
    }
    printf("\n");
}
```



```

        fprintf(fp, "\n");

        fclose(fp);

        return 0;
    }
    /*-----

```

【程序填空】

题目：任何一个大于 4 的偶数都可以表示成两个素数之和，输入一个偶数，为其寻找两个素数，这两个素数之和等于该偶数。
例如：16=3+13, 3 和 13 都是素数。

注意：请勿改动主函数 main() 中的其他内容。

-----*/

```

#include<stdio.h>
#include<math.h>
void main()
{
    void fun(int a);
    int a;
    do
    {
        printf("Please input a:");
        scanf("%d",&a);
    }while(a%2);
    fun(a);
}
void fun(int a)
{
    int b,c,d,y,i,j;
    for(i=3;i<=a/2;i=i+2)
    {
        /******SPACE*****/
        y=1;
        for(j=2;j<=sqrt(i);j++)
            if(i%j==0) y=0;
        if(y==1)
        {
            /******SPACE*****/
            d=a-i;

```

```

        for(j=2;j<=sqrt(d);j++)
            if(d%j==0) y=0;
        if(y==1)
            {b=i;c=d;printf("%d=%d+%d\n",a,b,c);break;}
    }
}
}

```

/*-----

【程序改错】

题目：下面程序中，sort 函数的功能是：将 n 个随机生成的无序整数从小到大排序。

请改正程序中的错误，使它能得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构。

-----*/

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#define MAX 10
void getarray(int [],int n);
void printArray(int [],int n);
void sort(int [],int n);

void main()
{
    int aa[MAX];
    srand((unsigned)time(NULL));    //初始化随机数时间序列
    getarray(aa,MAX);
    printf("\n\nThe Array Before sorted:\n");
    printArray(aa,MAX);
    sort(aa,MAX);
    printf("\n\nThe Array After sorted:\n");
    printArray(aa,MAX);
}

void getarray(int a[],int n)
{
    int i=0;

```

```

/*****FOUND*****/
    for(;i<n;i++)
    {
        a[i]=rand()%99+1;
    }

}
void sort(int a[],int n)
{
    int i, j, k, t;
    i=0;
    do
    {
        k=i;
/*****FOUND*****/
        j=i+1;
        while(j<n)
        {
            if(a[j]<a[k]) k=j;
            j++;
        }
/*****FOUND*****/
        if(a[k]<a[i])
        {
            t=a[i];a[i]=a[k];a[k]=t;
        }
        i++;
    }while(i<n-1);
}

void printArray(int a[], int n)
{
    int i=0;
    for (; i<n; i++)
    {
        printf("%5d", a[i]);
    }
    printf("\n");
}
/*****

```

题目：主函数 main()中由数字与字母组成的字符串 s 为测试数据，程序主要从字符串数组中找出最长的字符串（如果有多个长度相同且为最长的字符串，则约定第一个为最长的字符串）；然后判断最

长字符串中所包含的数字子串对应整数是否是素数。

例如：字符串 s 为"123ABC"、"23A48DE89"、"ABC3247"，则最长的字符串为"23A48DE89"，其数字子串对应整数为素数的整数为 23，89。

编写程序：

1. 编写函数 int MaxStr(char s[][80],int n),查找字符串数组 s 中 n 个字符串中最长的字符串，函数返回最长字符串所在的数组行的行下标。
2. 编写函数 int TestPrime(char s[],int b[]),判断最长字符串 s 中数字子串对应整数是否是素数，不同素数保存在数组 b 中，函数返回素数的个数。

注意：请勿改动主程序 main()中的任何语句。

*****/

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<conio.h>
```

```
#include<ctype.h>
```

```
#include<string.h>
```

```
#define N 80
```

```
int MaxStr(char s[][80],int n)
```

```
{
```

```
/******Program*****/
```

```
int i,max=0;
```

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

```
if(strlen(s[i])>strlen(s[max]))
```

```
max=i;
```

```
return max;
```

```
/****** End *****/
```

```
}
```

```
int TestPrime(char s[],int b[])
```

```
{
```

```
/******Program*****/
```

```
/* 判断最长字符串 s
```

```
中数字子串对应整数是否是素数，不同素数保存在数组 b 中，函
```

```
数返回素数的个数。*/
```

```
int i,j,k=0,flag,num=0;
```

```
for(i=0;s[i];i++)
```

```

{
    flag=1;
    if(isdigit(s[i]))
        num=num*10+s[i]-'0';
    if(isdigit(s[i+1]))
        continue;
    else
    {
        /*判断素数模块*/
        for(j=2;j<num;j++)
            if(!(num%j))
                break;
        if(j==num)
        {
            /*说明是素数，进行下面的操作*/
            for(j=0;j<k;j++)
                if(b[j]==num)
                    break;
            if(j==k)
                b[k++]=num;
        }
        /*判断素数模块*/
        num=0;
    }
}
return k;
/*****      End      *****/
}

int main()
{
    char s[][80]={"123ABC","23A48DE89","ABC3247"};
    int b[N]={0};
    int num=0,n=0,i=0;

    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("File open error\n");
        exit(0);
    }

    n=MaxStr(s,3);
    printf("The max string is: %s\n",s[n]);

```

```

    fprintf(fp,"The max string is: %s\n",s[n]);

    num=TestPrime(s[n],b);
    printf("The prime list is: ");
    fprintf(fp,"The prime list is: ");

    for(i=0;i<num;i++)
    {
        printf("%d  ",b[i]);
        fprintf(fp,"%d  ",b[i]);
    }

    printf("\n");
    fprintf(fp,"\n");

    fclose(fp);

    return 0;
}
/*-----

```

【程序填空】

题目：将字符串 s 中的所有数字字符顺序前移，其他字符顺序后移。
 例如，s 为 asd123fgh456，则处理后的新字符串应为
 123456asdfgh。

注意：请勿改动主函数 main() 中的其他内容。

```

-----*/

#include<stdio.h>
#include<string.h>
#include<ctype.h>
void main()
{
    void fun(char s[]);
    char s[80];
    printf("Please input the string:");
    scanf("%s",s);
    fun(s);
    printf("The result is %s\n",s);
}

void fun(char s[])

```

```

{
    int i,j,k;
    char p[80],t[80];
    j=k=0;
    for(i=0;s[i]!='\0';i++)
    {
        /******SPACE*****/
        if(isdigit(s[i])) p[j++]=s[i];
        else t[k++]=s[i];
    }
    for(i=0;i<j;i++)
        s[i]=p[i];
    for(i=0; i<k;i++)
        /******SPACE*****/
        s[j+i]=t[i];
    s[j+k]=0;
}

/*-----

```

【程序改错】

题目：下面程序的功能是：将一维字符数组中以空格为分隔符的单词转存到二维字符数组中，使二维数组的每一行以字符串的形式保存一个单词，然后将二维字符数组中的单词以每行一个的方式输出到屏幕上。其正确的输出是：

```

Microsoft
data
access
technologies

```

请改正程序中的错误，使它能得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构。

```

-----*/

#include <stdio.h>
#include <stdlib.h>
#define W_LEN 15    //单词长度，即单词中所允许包含字母的最大个数
#define W_COUNT 20  //字符串中所允许包含单词的最大个数

int changeStr(char [],char[][W_LEN]);

```

```

void main()
{
    char text[]="  Microsoft data  access technologies";
    char word[W_COUNT][W_LEN];
    int wcount=0;    //统计字符串中所含单词的个数
    int i;
    /*******FOUND*****/
    wcount=changeStr(text,word);

    for(i=0;i<wcount;i++)
    {
        puts(word[i]);
    }

}

int changeStr(char str[],char word[][W_LEN])
{
    int i=0,j=0,k=0,newword=1;
    do{
        if(str[k]==32)    //32 为空格字符的 ASCII 码
        {
            if(!newword)
            {
                newword=1;
                /*******FOUND*****/
                word[i][j]='\0';
                j=0;
                i++;
            }
            k++;
        }
        else
        {
            if(newword)    newword=0;
            word[i][j]=str[k];
            k++;
            j++;
        }
    }while(str[k]);
    /*******FOUND*****/
    if(str[k-1]==32)
        return i;
}

```



```

        else
        {
            word[i][j]='\0';
            return i+1;
        }
    }
}
/*****

```

题目：主函数 main() 中由数字与字母组成的字符串 s 和 t 为测试数据，程序首先将未在字符串 s 中出现、而在字符串 t 中出现的字符，生成一个新的字符串 a；然后计算字符串 a 中所有数字子串对应的整数，找出其平方数中有连续的三位数字就是该数本身的整数。

例如：当 s="AB148", t="E250A148C259G376" 时，字符串 a 为 "E250C259G376"。a 中数字子串 "250"，其对应整数平方数为 62500，250 便是所要找的一个满足条件的整数。

编写程序：

1. 编写函数 void GenStr(char s[], char t[], char a[])，将未在字符串 s 中出现、而在字符串 t 中出现的字符保存在字符数组 a 中。
2. 编写函数 int Calculate(char a[], int b[])，计算字符串 a 中数字子串，按上述要求找出所要的整数，结果保存在整数数组 b 中，函数返回整数个数。

注意：请勿改动主程序 main() 中的任何语句。

*****/

```

#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>
#define N 80

void GenStr(char s[], char t[], char a[])
{
/*****Program*****/
    int i=0, j=0, cd=0, flag;
    while(t[i])
    {
        flag=1;
        j=0;

```

```

while(s[j])
{
    if(s[j]==t[i])
    {
        flag=0;
        break;
    }
    j++;
}
if(flag)
    a[cd++]=t[i];
i++;
}
a[cd]=0;
/*****      End      *****/
}

```

```

int Calculate(char a[],int b[])
{
/*****Program*****/

    int k=0;
    int i=0,zcc=0,s=0,j;
    char zc[20],pf[20];
    long sum;
    while(a[i])
    {
        if(isdigit(a[i]))
        {
            zc[zcc++]=a[i];
            s=s*10+a[i]-'0';
        }
        if(isdigit(a[i])&&!isdigit(a[i+1]))
        {
            sum=s*s;
            zcc=0;
            while(sum)
            {
                pf[zcc++]=sum%10+48;
                sum=sum/10;
            }
            pf[zcc]=0;
            strrev(pf);
            for(j=0;j<(strlen(pf)-strlen(zc)+1);j++)

```

```

        {
            for(zcc=0;zc[zcc]&&(zc[zcc]==pf[j+zcc]);zcc++)
                ;
            if(!zc[zcc])
                b[k++]=s;

        }
    }
    if(!isdigit(a[i]))
    {
        sum=0;
        s=0;
        zcc=0;
        j=0;
    }
    i++;
}
return k;
/*****      End      *****/
}

```

```

int main()
{
    char s[]="AB148";
    char t[]="E250A148C259G376";
    char a[N];
    int b[N]={0};
    int num=0,i=0;

    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("File open error\n");
        exit(0);
    }

    GenStr(s,t,a);
    printf("The string a is: %s\n",a);
    fprintf(fp,"The string is: %s\n",a);

    num=Calculate(a,b);
    printf("The integer list is: ");
    fprintf(fp,"The integer list is: ");
    for(i=0;i<num;i++)

```

```

    {
        printf("%d ",b[i]);
        fprintf(fp,"%d ",b[i]);
    }

    printf("\n");
    fprintf(fp,"\n");

    fclose(fp);

    return 0;
}

```

/*-----

【程序填空】

题目：计算 x 数组中 N 个数的平均值（规定所有数均为正数），将数组中小于平均值的数据顺序移动到数组的前部，大于等于平均值的数据顺序移到 x 数组的后部。例如，有 10 个正数：47、30、32、40、6、17、45、15、48、26，其平均值为 30.5，移动后的输出为：30、6、17、15、26、47、32、40、45、48。

注意：请勿改动主函数 main() 中的其他内容。

-----*/

```

#include<stdio.h>
#include<stdlib.h>
#define N 10
void main()
{
    double fun(double x[]);
    int i; double x[N];
    for(i=0;i<N;i++)
    {
        x[i]=rand()%50;
        printf("%4.0f",x[i]);
    }
    printf("\n");
    printf("The average is:%f\n",fun(x));
    for(i=0;i<N;i++)

```

```

        printf("%5.0f",x[i]);
    printf("\n");
}
double fun(double x[])
{
    int i,j;
    double aver=0,y[N];
    for(i=0;i<N;i++)
        aver+=x[i];
    aver=aver/N;
    for(i=j=0;i<N;i++)
        if(x[i]<aver)
        {
            /*****SPACE*****/
            y[j]=x[i];x[i]=-1;j++;
        }
    i=0;
    while(i<N)
        /*****SPACE*****/
        {
            if(x[i]!=-1) y[j++]=x[i];
            i++;
        }
    for(i=0;i<N;i++) x[i]=y[i];
    return aver;
}
/*-----

```

【程序改错】

题目：下面程序中，函数 sort()的功能是将数组 s 中存放的英文段落中的所有单词取出以字典升序的顺序保存到数组 w 中。

测试数据：microsoft data access technologies.

运行结果：access data microsoft technologies.

请改正程序中的错误，使它能得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构。

```

-----*/
#include <stdio.h>
#include <string.h>
#include <ctype.h>

```

```

#define ROW 10
#define COL 20

/*****FOUND*****/
int sort(char s[],char[][COL]);

void main()
{
    char sent[]="microsoft data access technologies.";
    char word[ROW][COL];
    int i,n;
    n=sort(sent,word);

    for(i=0;i<n-1;i++)
        printf("%s ",word[i]);
    printf("%s\n",word[i]);
}

int sort(char s[],char w[][COL])
{
    int i=0,j=0,k=0,wordcount=0;
    char b[COL];
    while(s[i]){
        if(isalpha(s[i]))
/*****FOUND*****/
            w[k][j++]=s[i];
        else{
            w[k][j]=0;
            k++;
            j=0;
        }
        i++;
    }
    for(i=0;i<k-1;i++)
        for(j=0;j<k-1-i;j++)
            if(strcmp(w[j],w[j+1])>0){
                strcpy(b,w[j]);
/*****FOUND*****/
                strcpy(w[j],w[j+1]);
                strcpy(w[j+1],b);
            }
    wordcount=k;
    return wordcount;
}

```

```
/******
```

题目：主函数 main() 中由数字与字母组成的字符串 s 为测试数据，程序首先从字符串 s 尾部开始，按逆序把在其中出现的每相邻的两个字符，紧随其后重复出现一次，若字符串 s 中头部有剩余的单个字符也重复，生成新的字符串 t；然后计算字符串 t 中数字子串对应整数的各位数字之和。

例如：当 s="A48CD06"，则字符串 t 为"6060DCDC8484AA"，其数字子串对应整数的各位数字之和为 12 和 24。

编写程序：

1. 编写函数 void GenStr(char s[],char t[])，按上述规则生成字符串，结果保存在字符数组 t 中。
2. 编写函数 int Calculate(char t[],int b[])，计算字符串 t 中各个数字子串对应整数的各位数字之和，结果保存在整数数组 b 中，函数返回整数个数。

```
*****
```

注意：请勿改动主程序 main() 中的任何语句。

```
*****/
```

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<conio.h>
```

```
#include<ctype.h>
```

```
#include<string.h>
```

```
#define N 80
```

```
void GenStr(char s[],char t[])
```

```
{
```

```
/******Program*****
```

```
int i,k=0;
```

```
char ls[20];
```

```
t[0]=0;
```

```
for(i=strlen(s)-1;i>=0;i--)
```

```
{
```

```
if(i!=0&&((isalpha(s[i])==isalpha(s[i-1]))||(isdigit(s[i])==isdigit(s[i-1])))) /*未到最前面一个字符并且当前字符与下面要处理的字符属于同一个类型*/
```

```
ls[k++]=s[i];
```

```
if(i!=0&&((isalpha(s[i])!=isalpha(s[i-1]))||(isdigit(s[i])!=isdigit(s[i-1])))) /*未到最前面一个字符并且当前字符与下面一个要处理的字符不一样*/
```

```
{
```

```

        ls[k++]=s[i];
        ls[k]=0;
        strcat(t,ls);
        strcat(t,ls);
        k=0;
    }
    if(i==0&&((isalpha(s[i])==isalpha(s[i+1]))||(isdigit(s[i])==isdigit(s[i+1]))))    /*已经
到了最前面一个字符并且当前字符同刚刚处理的字符的类型一样*/
    {
        ls[k++]=s[i];
        ls[k]=0;
        strcat(t,ls);
        strcat(t,ls);
    }
    if(i==0&&((isalpha(s[i])!=isalpha(s[i+1]))||(isdigit(s[i])!=isdigit(s[i+1]))))    /*已经
到了最前面一个字符并且当前字符同刚刚处理的字符的类型不一样也
就是说这是个新字符*/
    {
        ls[0]=s[i];
        ls[1]=0;
        strcat(t,ls);
        strcat(t,ls);
    }
}
/***** End *****/
}

```

```

int Calculate(char t[],int b[])
{
/*****Program*****/
int i,flag=0,k=0,s=0;
for(i=0;t[i];i++)
{
    if(isdigit(t[i])&&!flag)    /*到达了数字字符开始的时候*/
    {
        s=s+t[i]-'0';
        if(isalpha(t[i+1]))
            b[k++]=s;
    }
    if(isdigit(t[i])&&flag)    /*数字字符的中间部分*/
    {
        s=s+t[i]-'0';
        if(isalpha(t[i+1]))
            b[k++]=s;
    }
}

```



```

    }
    if(isalpha(t[i]))                /*假如到了字符需要给下一次做准备*/
    {
        flag=0;
        s=0;
    }
    if(isdigit(t[i]))
    {
        flag=1;
    }
    }
    return k;
/*****      End      *****/
}

```

```

int main()
{
    char s[]="A48CD06";
    char t[N];
    int b[N]={0};
    int num=0,i=0;

    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("File open error\n");
        exit(0);
    }

    GenStr(s,t);
    printf("The string t is: %s\n",t);
    fprintf(fp,"The string t is: %s\n",t);

    num= Calculate(t,b);
    printf("Sum of the digits of each integer is: ");
    fprintf(fp,"Sum of the digits of each integer is: ");

    for(i=0;i<num;i++)
    {
        printf("%d ",b[i]);
        fprintf(fp,"%d ",b[i]);
    }

    printf("\n");
}

```

```

        fprintf(fp, "\n");

        fclose(fp);

        return 0;
}

```

七、程序填空 共 1 题 （共计 20 分）

第 1 题 （20.0 分）

/*-----

【程序填空】

题目：建立一个 $N \times N$ 的矩阵，矩阵元素的构成规律是：最外层元素的值全部为 1；从外向内第 2 层元素的值全部为 2；第 3 层元素的值全部为 3，……依此类推。例如，若 $N=5$ ，则生成的矩阵为：

```

1  1  1  1  1
1  2  2  2  1
1  2  3  2  1
1  2  2  2  1
1  1  1  1  1

```

注意：请勿改动主函数 `main()` 中的其他内容。

-----*/

```

#include<stdio.h>
#define N 7
void main()
{
    void fun(int x[][N],int n);
    int x[N][N]={0},i,j;
    fun(x,N);
    printf("The result is:\n");
    for(i=0;i<N;i++)
    {
        for(j=0;j<N;j++)
            printf("%4d",x[i][j]);
        printf("\n");
    }
}

```

```

void fun(int x[][N],int n)
{
    int i,j,k,m;
    if(n%2==0) m=n/2;
    else m=n/2+1;           // m 表示不同字符数字所在的圈数
    for(i=0;i<m;i++)
    {
        /*****SPACE*****/
        for(j=i;j<n-i;j++)
            x[i][j]=x[n-i-1][j]=i+1;
        for(k=i+1;k<n-i;k++)
        /*****SPACE*****/
            x[k][i]=x[k][n-i-1]=i+1;
    }
}

```

八、程序改错 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

```

/*-----
【程序改错】
-----

```

题目：下面程序的功能是：首先随机产生 N 个 2 至 3 位的整数并将其存放于数组 s 中，然后从 s 中找出所有的互质数对(所谓的互质数对是指两个数最大公约数是 1)，将所找出的互质数对存放于二维数组 t 中。

请改正程序中的错误，使它能得出正确的结果。

注意：不得增行或删行，也不得更改程序的结构。

```

-----*/

```

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

```

```

#define N 5

```

```

int find(int[],int[][2],int); //查找互质数对

```

```

int huzhi(int,int);    //判断两个正整数是否互质
void main()
{
    int s[N],t[N*(N-1)/2][2],n=0,i=0;

    srand((unsigned)time(NULL));    //初始化随机数时间序列
    for(i=0;i<N;i++)    //随机产生 N 个 2 至 3 位正整数
    {
        s[i]=rand()%990+10;
        printf("%6d",s[i]);
    }
    printf("\n");

    n=find(s,t,N);

    for(i=0;i<n;i++)
        printf("%6d,%6d\n",t[i][0],t[i][1]);
}

/*****FOUND*****/
int find(int s[],int t[][2],int n){
    int i,j,k=0;
    for(i=0;i<n-1;i++)
/*****FOUND*****/
        for(j=0;j<n;j++)
            if(huzhi(s[i],s[j])){
                t[k][0]=s[i];
                t[k][1]=s[j];
                k++;
            }
    return k;
}

int huzhi(int m,int n){
    int r;
    if(m<n){
        r=m;m=n;n=r;}
    r=0;
    do{
        r=m%n;
        m=n;
        n=r;
/*****FOUND*****/
    }while(!r);
}

```

```
    if(m==1)
        return 1;
    else
        return 0;
}
```

九、程序设计 共 1 题 （共计 30 分）

第 1 题 （30.0 分）

/******

题目：主函数 main()中由数字与字母组成的字符串 s 为测试数据，程序
首先把第 1 到第 m 个字符平移到字符串的最后，第 m+1 到最后的字
符移到字符串的前部，生成新的字符串 s；然后计算字符串 s 中所
有相邻两个数字子串对应整数的最大公约数。

例如：字符串 s 为"abc12efg20mf30"，m 的值为 3，则移动后，新字符串 s
为"12efg20mf30abc"。

编写程序：

1. 编写函数 void GenStr(char s[],int m)，按上述要求生成新的
字符串，结果仍然保存在字符数组 s 中。
2. 编写函数 int Calculate(char s[],int b[]),计算字符串 s 中相
邻两个数字子串对应整数的最大公约数，结果保存在整数数组 b
中，函数返回整数个数。

注意：请勿改动主程序 main()中的任何语句。

*****/

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>
#define N 80
```

```

void GenStr(char s[],int m)
{
/*****Program*****/

/***** End *****/
}

int Calculate(char s[],int b[])
{
/*****Program*****/

/***** End *****/
}

int main()
{
    char s[]="abc12efg20mf30";
    int b[N]={0};
    int num=0,i=0;

    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("File open error\n");
        exit(0);
    }

    GenStr(s,3);
    printf("The string s is: %s\n",s);
    fprintf(fp,"The string s is: %s\n",s);

    num= Calculate(s,b);
    printf("The max factor is: ");
    fprintf(fp,"The max factor is: ");

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

```

```

        printf("%d ",b[i]);
        fprintf(fp,"%d ",b[i]);
    }

    printf("\n");
    fprintf(fp,"\n");

    fclose(fp);
    getch();
    return 0;
}

/*-----
【程序填空】
-----*/

```

题目：找出矩阵 x 前 n 行中的最大整数，用该整数所生产的降序数将其替换。降序数是指该数的高位数字的值不小于低位数字的值。例如，若矩阵中的最大整数是 1024，则用 4210 替换 1024。

例如，矩阵 x 为：

456	44	674
789	324	1024
213	345	537

则替换后的矩阵为：

456	44	674
789	324	4210
213	345	537

注意：请勿改动主函数 main() 中的其他内容。

```

-----*/

#include<stdio.h>
void main()
{
    void matrix_change(int x[][3],int n);
    int a[3][3]={456,44,674,789,324,1024,213,345,537},i,j;
    printf("The result is:\n");
    matrix_change(a,2);
    for(i=0;i<3;i++)
    {

```

```

        for(j=0;j<3;j++) printf("%5d",a[i][j]);
        printf("\n");
    }
}
void matrix_change(int x[][3],int n)
{
    int i,j,k,t[5]={0},max,maxi,maxj,v;
    max=x[0][0];maxi=maxj=0;
    for(i=0;i<n;i++)
        for(j=0;j<3;j++)
            if(x[i][j]>max) {max=x[i][j];maxi=i;maxj=j;}
    k=0;
    while(max>0)
    /*****SPACE*****/
    {   t[k++]=max%10;max=max/10;   }
    for(i=0;i<k;i++)
    /*****SPACE*****/
        for(j=0; j<k-i-1;j++)
            if(t[j]<t[j+1]) {v=t[j];t[j]=t[j+1];t[j+1]=v;}
    for(i=0;i<k;i++)
        max=max*10+t[i];
    x[maxi][maxj]=max;
}

```

/*-----
【程序改错】

题目：下面程序的功能是：验证是否存在两个大于 2 而小于 n 的素数
 p1 与 p2，使得 $n=p1+p2$ 。(2012≤n≤2016)

请改正程序中的错误，使它能得出正确的结果。

 注意：不得增行或删行，也不得更改程序的结构。
 -----*/

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

```

int prime(int);
int find(int,int[]);    //寻找符合条件的素数对
void main()
{

```



```

int n,p[2];
for(n=2012;n<=2016;n++){
    if(find(n,p))
        printf("%d=%d+%d\n",n,p[0],p[1]);
    else
        printf("Not Exist m and n make %d=m+n\n",n);
}
}

```

/******FOUND******/

```

int find(int n,int p[])
{
    int n1=3,n2,flag=0;

    while(n1<=n/2 &&!flag)
    {
        if(prime(n1))    n2=n-n1;
/******FOUND******/
        if(prime(n2==1))
        {
            p[0]=n1;
            p[1]=n2;
            flag=1;
        }
        n1=n1+2;
    }
    return flag;
}

```

int prime(int n) //判断一个正整数 n 是否为素数

```

{
    int i=2;
/******FOUND******/
    while(i<=sqrt(n)&& n%i!=0) i++;

    if(i>sqrt(n))    return 1;
    return 0;
}
/******

```

题目：主函数 main()中由数字与字母组成的字符串 s 为测试数据，程序首先删除字符串 s 中下标为奇数位置上指定字符 ch，生成新的字符串 s；然后计算字符串 s 中各个数字子串对应整数的反序数。

例如：字符串 s 为："abc12efg24mf36"，指定字符 ch 为'f'，删除下标为奇数位置上字符'f'后，生成的新的字符串 s 为"abc12efg24m36"，数字子串"12"对应整数 12 的反序数为 21。

编写程序：

1. 编写函数 void GenStr(char s[],char ch)，按上述要求删除指定字符 ch，结果仍然保存在字符数组 s 中。
2. 编写函数 int Calculate(char s[],int b[])，计算字符串 s 中数字子串对应整数的反序数，结果保存在整数数组 b 中，函数返回整数个数。

注意：请勿改动主程序 main()中的任何语句。

*****/

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<conio.h>
```

```
#include<ctype.h>
```

```
#include<string.h>
```

```
#define N 80
```

```
void GenStr(char s[],char ch)
```

```
{
```

```
/******Program*****/
```

```
int i,k,j;
```

```
for(i=0,k=strlen(s);i<k;i++)
```

```
{
```

```
if((s[i]==ch)&&(i%2))
```

```
{
```

```
for(j=i;j<k;j++)
```

```
s[j]=s[j+1];
```

```
k--;
```

```
}
```

```
}
```

```
puts(s);
```

```
/******End*****/
```

```
}
```

```
int Calculate(char s[],int b[])
```

```
{
```

```
/******Program*****/
```

```

int k=0,i,j,num=0,t=0;;
for(i=0;s[i];i++)
{
    if(isdigit(s[i]))
        num=num*10+s[i]-'0';
    if(isdigit(s[i])&&!isdigit(s[i+1]))
    {
        while(num)
        {
            t=t*10+num%10;
            num=num/10;
        }
        b[k++]=t;
        t=0;
    }
}
return k;
/*****End*****/
}

```

```

int main()
{
    char s[]="abc12efg24mf36";
    int b[N]={0};
    int num=0,i=0;

    FILE *fp;
    if((fp=fopen("DATA.TXT","w"))==NULL)
    {
        printf("File open error\n");
        exit(0);
    }

    GenStr(s,'f');
    printf("The string s is: %s\n",s);
    fprintf(fp,"The string s is: %s\n",s);

    num= Calculate(s,b);
    printf("The integer list is: ");
    fprintf(fp,"The integer list is: ");

    for(i=0;i<num;i++)
    {
        printf("%d ",b[i]);
    }
}

```

```
        fprintf(fp,"%d ",b[i]);
    }

    printf("\n");
    fprintf(fp,"\n");

    fclose(fp);

    return 0;
}
```