转译：

\b 退格

\f 换页

\n 换行

\r 回车

\t 水平制表

\v 垂直制表

\\ 反斜杠

\? 问号字符

\‘ 单引号字符

\" 双引号字符

\0 空字符

### 1.求int a[]={12,43,56,14,78,16,50,26,30,40};的最大值；

实现方法一：

int a[]={12,43,56,14,78,16,50,26,30,40};

int max=a[0];

int i=0;

int len = sizeof(a)/sizeof(int);

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

if(a[i]>=max){

max=a[i];

}

}

lr\_output\_message("max=%d",max);

return 0;

实现方法二（函数）：

Action()

{

// 求int a[]={12,43,56,14,78,16,50,26,30,40};的最大值

int a[]={12,43,56,14,78,16,50,26,30,40};

int len = sizeof(a)/sizeof(int);

int max = getMax(a,len);

lr\_output\_message("max=%d",max);

return 0;

}

int getMax(int \*ptr, int len){

int \*p=ptr;

int max1=\*p;

for(;p<(ptr+len);p++){

if(\*p>=max1){

max1=\*p;

}

}

return max1;

}

---------------------------------------------------------------------------------------------

### 2.试写一函数，求n的阶乘

long fac(unsigned n)

{

if (n == 0 || n == 1)

{

return 1;

}

else

{

return n \* fac(n - 1);

}

}

Action()

{

int a = 4;

fac(a);

lr\_output\_message("%d",fac(a));

return 0;

}

/////////////

Action()

{

//写一个函数:求x的阶乘int getFactorial(int n);

long int fac=getFactorial(3);

lr\_output\_message("fac=%ld",fac);

}

long int getFactorial(long int n){

long int i=2;

long int fac=1;

if(n==0||n==1){

fac=1;

}

while(i<=n){

fac=fac \* i;

i++;

}

lr\_output\_message("%ld",fac);

return fac;

}

---------------------------------------------------------------------------------

### 3.试写一函数，求int a[]={12,43,56,14,78,16,50,26,30,40};的中位数

int \* sort(int \*x,int n)

{

int j, k, h, t;

lr\_output\_message("len=%d",n);

for (h=n-1; h>0; h=k) /\*循环到没有比较范围\*/

{

for (j=0, k=0; j<h; j++) /\*每次预置k=0，循环扫描后更新k\*/

{

if (\*(x+j) > \*(x+j+1)) /\*大的放在后面，小的放到前面\*/

{

t = \*(x+j);

\*(x+j) = \*(x+j+1);

\*(x+j+1) = t; /\*完成交换\*/

k = j; /\*保存最后下沉的位置。这样k后面的都是排序排好了的。\*/

}

}

}

if(n % 2 == 0){

lr\_output\_message("中位数=%lg\n",(x[n/2] + x[n/2 - 1]) / 2.0);

}else{

lr\_output\_message("中位数=%lg\n", x[n/2]\*1.0);

}

return 0;

}

Action()

{

int a[] ={12,43,56,14,78,16,50,26,30,40};

int i;

i= sizeof(a)/sizeof(int);

sort(a,i);

lr\_output\_message("len=%d",i);

for (i=0;i<3;i++) {

lr\_output\_message("%d",a[i]);

}

return 0;

}

////////////////

实现方法一：

Action()

{

//求int a[]={12,43,56,14,78,16,50,26,30,40};的中位数

int a[]={12,43,56,14,78,16,50,26,30,40};

int len=sizeof(a)/sizeof(int);

int temp;

int i=0,j=0;

int k=0;

int mid;

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

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

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

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

for(k=0;k<len;k++){

lr\_output\_message("%d",a[k]);

}

if(len %2==1){

mid=(len+1)/2;

lr\_output\_message("数组的中位数是:%d",a[mid]);

}

if(len %2 ==0)

{

lr\_output\_message("数组的中位数是:%d,%d",a[(len/2)-1],a[(len/2)-1]);

}

}

实现方法二（函数）：

int getMid(int \*ptr, int len);

Action()

{

//求int a[]={12,43,56,14,78,16,50,26,30,40};的中位数

int a[]={12,43,56,14,78,16,50,26,30,40};

int len=sizeof(a)/sizeof(int);

getMid(a,len);

return 0;

}

int getMid(int \*ptr, int len){

int \*p=ptr;

int \*p1=p+1;

int temp;

lr\_output\_message("-----------%d",len);

for(;p<(ptr +len-1);p++){

for(p1=p+1;p1<(ptr + len);p1++){

if(\*p>\*p1){

temp=\*p;

\*p=\*p1;

\*p1=temp;

}

}

}

for(p=ptr;p<(ptr+len);p++){

lr\_output\_message("%d",\*p);

}

if(len %2==1){

p=ptr;

lr\_output\_message("数组的中位数是:%d",\*(p+((len-1)/2)));

}

if(len %2 ==0)

{ p=ptr;

lr\_output\_message("数组的中位数是:%d",(\*(p+((len/2)-1)) + \*(p+((len/2))))/2);

}

return 0;

}

------------------------------------------------------------------------------------------------

### 4.试写一函数，求100以内所有素数之和

int sumss(int n){

int i,j;

int sum=1;//素数之和默认为1

for(i=2;i<=n;i++) //从2开始计算

{ for(j=2;j<=i-1;j++)

{

if(i%j==0)

break; }

if(j>=i)

sum+=i;

}

lr\_output\_message("%d\n",sum);

return sum;

}

Action()

{

int a=100;

lr\_output\_message("１００内的所有素数之合%d\n",sumss(a));

return 0;

}

///////////////

void continueTest4();

Action()

{

//试写一函数，求100以内所有素数之和

continueTest4();

return 0;

}

void continueTest4(){

int i,j;

int sum=0;

for(i=2;i<=100;i++){

int t=1;

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

if(i%j==0){

t=0;

break;

}

}

if(t==1)

{

//lr\_output\_message("%d",i);

sum=sum+i;

}

}

lr\_output\_message("continueTest4 sum=%d",sum);

}

-------------------------------------------------------------------

### 5.重写strcpy函数

char \*my\_strcpy(char \*dest, const char \*src)

{

char \*old = dest;

while (\*dest = \*src)

{

dest++;

src++;

}

return old;

}

Action()

{

char arr1[] = "abcdefg";

char arr2[64];

my\_strcpy(arr2, arr1);

lr\_output\_message("the string of copy is:>%s\n", arr2);

return 0;

}

--------------------------------------------------------------------

### 6.重写strcat函数

char mycat(char \* ptr,const char \* src);

Action()

{

char str1[] = "China";

char str2[] = "Zhejiang";

mycat(str1, str2);

lr\_output\_message("%s",str1);

return 0;

}

char mycat(char \* ptr,const char \* src){

if(src == 0 || ptr == 0){

lr\_output\_message("您输入的参数错误");

return -1;

}

while(\*ptr!='\0' ){

\*ptr++ ;

}

while(\*src!= '\0'){

\*ptr++ = \*src++;

}

lr\_output\_message("ptr=%s",ptr);

\*ptr = '\0';

return \*ptr;

}

-------------------------------------------------------------------------------------

### 7. 重写srtrchr函数，实现查找字符串中指定字符最后一次出现时，所有的字符输出，将其中指定字符串的小写字母变成对应的大写字母并输出。如字符串“testroad”,指定字符串为't'，则输出"Troad"

char \* my\_strrchr(char \*str,int ch)

{

int count=0;

while(\*str!='\0')

{

count++;

str++;

}

str--;

while(count)

{

if(\*str!=ch)

{

str--;

count--;

}

else

return str;

}

printf("未找到该字符。\n");

return 0;

}

Action()

{

char ch='t';

char \*src="testroad";

char const \*ret;

ret=my\_strrchr(src,ch);

lr\_output\_message("%s\n",ret);

return 0;

}

//////////////////

//函数声明

char \*strchrTest( char \*ptr, char c);

Action()

{

char str[]={"testroad"};

char c='t';

//strchrTest调用函数

lr\_output\_message("%s",strchrTest(str,c));

return 0;

}

char \*strchrTest(char \*ptr, char c){

char \*p=ptr;

char \*p1=ptr;

if(ptr !=NULL){ //异常处理

while(\*ptr !='\0'){ //移动指针到字符串尾

ptr++;

}

while(ptr !=p1){ //逆向查找指定字符

if(\*ptr == c)

{

p=ptr;

break;

}

ptr--;

}

}

//实现大写到小写的转化；

if((\*p>='A')&&(\*p<='Z')){

}

else if ((\*p>='a')&&(\*p<='z')){

\*p=\*p - 32;

}else{

lr\_output\_message("%c不是字母",c);

}

//lr\_output\_message("%s",ptr);

return p;

}

-----------------------------------------------------------------------------

### 8、定义一个函数，实现对字符串做如下操作：当字符为字母时，大小写互换，当字符为数字时，原样输出，当有其它字符出现时，结束操作，返回已处理的字符串

//函数声明

char \*strTest2( char \*ptr,int len);

Action()

{

char str[]={"tEst234%^road"};

int len = sizeof(str)/sizeof(char);

//strTest2调用函数

lr\_output\_message("%s",strTest2(str,len));

return 0;

}

char \*strTest2(char \*ptr,int len)

{

char \*p=(char \*)malloc(len);

char \*pt=p;

//lr\_output\_message("%s",ptr);

if(ptr !=NULL)

{

while(1)

{

\*p=\*ptr;

if((\*ptr>='A')&&(\*ptr<='Z')){

\*p=\*p + 32;

}

else if((\*ptr>='a')&&(\*ptr<='z')){

\*p=\*p - 32;

}

else if(\*ptr>='0' && \*ptr <='9'){

\*p = \*ptr;

}

else

{

\*p='\0';

break;

}

p++;

ptr++;

}

}

//lr\_output\_message("%s",pt);

return pt;

}

<<<<<<< HEAD

==============================================================

### 9.截取数组,逗号分隔符

extern char \* strtok(char \* string, const char \* delimiters );

char path[] = "www.baidu.com,www.sina.com,www.cctv.com";

char \* url1;

char \* url2;

char \* url3;

char url[1024];

char url4[1024]="http://";

char separators[] = ",";

url1 = (char \*)strtok(path, separators); //取path第一个逗号

url2 = (char \*)strtok(NULL, separators);//取第二个

url3 = (char \*)strtok(NULL, separators); //取第三个

lr\_output\_message ("url=%s , ur2=%s ,ur3=%s ", url1,url2,url3 );

=======

===========================================

### 10截取数组匹配"x"个字符

char \*str123(const char \*str, int c)

{

char \* p = NULL;// 初始化p

if(str == NULL) //判断传入字符串为空时,返回null

{

return NULL;

}

while(\*str != '\0') //条件 str不为0

{

if(\*str == (char)c) //取c的值匹配字符串

{

p = (char \*)str;//将str赋值给p

}

str++; //自增取最后一个值

// lr\_output\_message("ssss=%s",str);

}

return p;

}

================================================================

### 11取航班中最便宜价格

char \* lessOutFight(char \* outFight){

int count = lr\_paramarr\_len("outFight");

char ch[] =";";

int i =1 ,minpos=0;

int mymin;

int \* cost;

int p[10];

//char \* lesOutFight[10];

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

// while(i){

// }

cost=(int \*) strtok(lr\_paramarr\_idx(outFight,i), ch);

lr\_output\_message("cost=%s",cost);

cost =(int \*)strtok(NULL, ch);

lr\_output\_message("cost=%d",cost);

p[i]= atoi(cost);

lr\_output\_message("p1=%d",p[i]);

}

mymin=p[10];

i=1;

while(i<=count){

if(p[i]<mymin){

mymin = p[i];

minpos = i;

lr\_output\_message("min=%d",mymin);

lr\_output\_message("pos=%d,i=%d",minpos,i);

}

i++;

}

lr\_output\_message("min=%d",mymin);

lr\_output\_message("pos=%d",minpos);

return lr\_paramarr\_idx(outFight,minpos);

}

=============================================================

### 12用户名随机生成

int number =rand()%3+1;

char username[1024]="xwb";

char \* daterand;

lr\_save\_datetime("%Y%m%d%H%M%S", TIME\_NOW + ONE\_HOUR+number , "date");

lr\_output\_message("depnumber=%s",lr\_eval\_string("{date}"));

daterand = lr\_eval\_string("{date}");

strcat(username,daterand);

lr\_output\_message ("nnn= %s", username);