



此刻打盹，你将做梦

此刻学习，你将圆梦

请输入：

```
int secondMaxItem(int * arr, int n)
{
//上面代码不要动

//your code...
int i;
int flag_max=0,flag;

//找出最大的数
for( i = 0; i < n ; i++ )
{
    if( arr[flag_max] < arr[i] )
    {
```

3、试题名称：求字符串中数字的和

题目描述：

已知，传入一个由数字字符组成的字符数组，将每个字符表示的数字求和后，作为返回值。  
//str是传入的数字字符串  
//返回值是数字之和  
//传入:@ “12345”  
//返回:15 (即1+2+3+4+5)

温馨提示：

方法名称：

```
int sumOfNumbersInString(const char * str)
```

请输入：

```
int sumOfNumbersInString(const char * str)
{
//上面代码不要动

//your code...
int sum=0;
int i=0;
while(str[i]!='\0')
{
    sum += str[i]-'0';
    i++;
}
return sum;
```

4、试题名称：返回字符串中字母的个数

题目描述:	统计一个字符串中英语字母的个数 //str字符串 //返回值是字母个数
温馨提示:	
方法名称:	int numberOfSubtitlesInString(const char * str)
请输入:	<pre> int numberOfSubtitlesInString(const char * str) { //上面代码不要动  //your code... int count =0; int i =0; // char *s =(char *)str; char *s = (char *) malloc(sizeof(char)); strcpy(s,str);  while( s[i]!='\0' ) { </pre>

#### 5、试题名称：返回数组中所有数的最小公倍数

题目描述:	已知，数组中所有的数都是正整数，返回这些数的最小公倍数 //arr是数组的首元素地址 //n是数组的元素个数 //返回值是数组中所有数的最小公倍数
温馨提示:	
方法名称:	int minCommonMultiple(int * arr, int n)
请输入:	<pre> int minCommonMultiple(int * arr, int n) { //上面代码不要动  //your code... int i,j,max=arr[0]; int reback;  //找到最大的数 for(i = 0; i &lt; n; i++ ) { if(max &lt; arr[i]) max = arr[i]; </pre>

6、试题名称：求字符串中的单词个数

<div> <div>题目描述：</div> <div>           传入一个字符串，已知字符串中连续的字母称为字符，除字母外有其他字符，返回字符串中的单词个数。            //str 传入的字符串            //返回值是单词的个数         </div> </div>	
<div> <div>温馨提示：</div> </div>	
<div> <div>方法名称：</div> </div>	<div>           int numOfWorksInString(const char * str)         </div>
<div> <div>请输入：</div> </div>	<div> <pre> int numOfWorksInString(const char * str) { //上面代码不要动  //your code... int count = 0; char *word; char *sep = " ,./;:'\"\\!@#\$\$%^*1234567890"; char *array = (char *) malloc(sizeof(char)); strcpy(array,str);  for( word = strtok(array,sep); word; word = strtok(NULL,sep)) { </pre> </div>

7、试题名称：返回字符串中数字的和

<div> <div>题目描述：</div> <div>           已知输入的字符串中有多个数字，返回数字之和            //str 是字符串            //返回值是数字之和            //传入 “a12b34c56”            //返回 102 （即12+34+56之和）         </div> </div>	
<div> <div>温馨提示：</div> </div>	
<div> <div>方法名称：</div> </div>	<div>           int sumOfNumbersInString(const char * str)         </div>
<div> <div>请输入：</div> </div>	

```
int sumOfNumbersInString(const char * str)
{
//上面代码不要动

//your code...
int sum = 0;
int i =0;
//char *s =(char *)str;
char *s =(char *) malloc(sizeof(char));
strcpy(s,str);

while( s[i]!='\0' )
{
```

上一项