

# 清华大学本科生考试试题专用纸

考试课程 计算机语言与程序设计 ( 卷) 2011 年 1 月 14 日

姓名                      班级                      学号                     

大题	题号	一	二	三	四	总分
	分数	30 分	30 分	35 分	5 分	
成绩						

## 试卷提交说明

- 1、在考卷上填写你的姓名、学号;
- 2、按要求编程, 调试运行, 将每道题的结果考屏, 粘贴在考卷对应的题后;
- 3、交卷时, 提交一个压缩文件包, 包括:
  - a) 文件名  
学号姓名, 比如: 2010123456 张三.rar
  - b) 文件包内容
    - ✧ 每道题的 c 工程文件夹;
    - ✧ 考卷。

一、字符串处理。编制一个程序, 要求实现如下功能:

1. 主函数循环运行, 当且仅当输入 '@' 时, 程序结束运行。 (10 分)
2. 函数 input() (10 分)

从键盘输入两个由数字组成的字符串 (每个字符串的长度不超过 10 个字符, 无空格), 并返回给主函数;

3. 函数 char \*interlaced(char\*p1,char \*p2) (10 分)

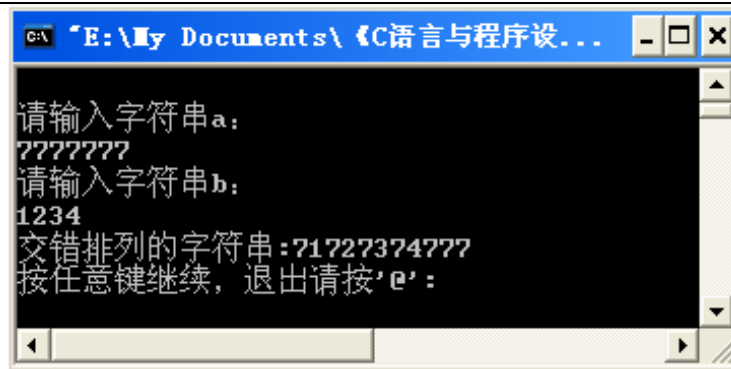
从字符串 p1 头部开始, 将 p1、p2 两字符串的数字, 依次交错地排成一个新的数字字符串 c, 并通过 return 返回给主函数。

举例如下:

- a) 输入字符串 a 是 "1234", b 是 "5678900", 则新的字符串 c 是:



- b) 输入字符串 a 是 "7777777", b 是 "1234", 则新字符串 c 是:



```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
void main()
{
    void input(char *p);
    char *interlaced(char *p1,char *p2);
    char str1[10],str2[10],*s1=str1,*s2=str2,*p;
    while(1)
    {
        printf("请输入字符串 a:\n");
        input(s1);
        printf("请输入字符串 b:\n");
        input(s2);
        printf("新的交错字符串是: \n");
        p=interlaced(s1,s2);
        printf("%s\n",p);
        printf("按任意键继续, 退出请按'@\n");
        if(getchar()=='@')
            break;
        fflush(stdin);
    }
}
void input(char *p)
{
    gets(p);
}
char *interlaced(char *p1,char *p2)
{
    int l1,l2,i;
    char *p;
    p=(char *)malloc(sizeof(char)*20); //必须申请 malloc 空间, 否则生成的 p 传不到主函数
    l1=strlen(p1); //!!!!
    l2=strlen(p2);
    if(l1>=l2)
    {
        for(i=0;i<l2;i++)
        {
            *(p+2*i)=*(p1+i);/*printf("%c",*(p+2*i))*/;
            *(p+2*i+1)=*(p2+i);/*printf("%c",*(p+2*i+1))*/;
        }
    }
}
```

```

    }
    for(i=l2;i<l1;i++)
    {
        *(p+l2+i)=*(p1+i);/*printf("%c",*(p+l2+i));*/
        *(p+l2+i)='\0';
    }
    if(l1<l2)
    {
        for(i=0;i<l1;i++)
        {
            *(p+2*i)=*(p1+i);/*printf("%c",*(p+2*i));*/
            *(p+2*i+1)=*(p2+i);/*printf("%c",*(p+2*i+1));*/
        }
        for(i=l1;i<l2;i++)
        {
            *(p+l1+i)=*(p2+i);/*printf("%c",*(p+l1+i));*/
            *(p+l1+i)='\0';
        }
    }
    return(p);
}

```

//第二次做的源代码

```

#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>
void main()
{
    char *input();
    char *interlaced(char *p1,char *p2);
    char str1[10],str2[10],str[20],*s1=str1,*s2=str2,*s=str;
    do
    {
        fflush(stdin);
        printf("请输入字符串 a\n");
        s1=input();
        printf("请输入字符串 b\n");
        s2=input();
        s=interlaced(s1,s2);
        printf("交错排列的字符串: \n");
        printf("%s\n",s);
        printf("按任意键继续, 退出请按@\n");
    }while(getchar()!='@');
}
char *input()
{
    char *p;
    p=(char *)malloc(sizeof(char)*10);
    gets(p);
    return(p);
}
char *interlaced(char *p1,char *p2)
{
    char *p;
    int i,l1,l2;
    p=(char *)malloc(sizeof(char)*20);
    l1=strlen(p1);

```

```
l2=strlen(p2);
if(l1>=l2)
{
    for(i=0;i<l2;i++)
    {
        *(p+2*i)=*(p1+i);
        *(p+2*i+1)=*(p2+i);
    }
    for(i=l2;i<l1;i++)
        *(p+l2+i)=*(p1+i);
}
if(l1<l2)
{
    for(i=0;i<l1;i++)
    {
        *(p+2*i)=*(p1+i);
        *(p+2*i+1)=*(p2+i);
    }
    for(i=l1;i<l2;i++)
        *(p+l1+i)=*(p2+i);
}
*(p+l1+l2)='\0';
return(p);
}
```

**二、链表编程。**程序功能如同第一题，但用链表实现字符串的存储及交错排列。其中：

1. **主函数循环运行**，读入字符串、生成链表、交互排列后输出，当且仅当输入'@'时，程序结束运行。(5分)

- ## 2. 函数 insert(char \*p) (10 分)

将从键盘输入的字符串 p 存储成链表，并将头指针通过 return 返回给主函数；

- ### 3. 函数 list(head) (10 分)

将 head 指向的链表输出到屏幕上。

4. 函数 `interlaced(p1, p2)` (5 分)

从 p1 头部开始，将 p1、p2 两字符串链表的各位数字，依次交错地排成一个新的数字字符串链表，并把头指针通过 return 返回给主函数。

举例如下：

a) 输入字符串 a 是"1234", b 是"5678900", 则链表 a、b、c 分别是:

```

C:\E:\My Documents\《C语言与程序设计基础... - [X]
请输入字符串a:
1234
字符串链表是:
1->2->3->4->NULL

请输入字符串b:
5678900
字符串链表是:
5->6->7->8->9->0->0->NULL
交错排列的字符串:
1->5->2->6->3->7->4->8->9->0->0->NULL
按任意键继续, 退出请按'e':

```

b) 输入字符串 a 是 "7777777", b 是 "1234", 则链表 a、b、c 分别是:

```
C:\ E:\My Documents\《C语言与程序设计基础》\10教...
请输入字符串a:
???????
字符串链表是:
7->7->7->7->7->7->NULL

请输入字符串b:
1234
字符串链表是:
1->2->3->4->NULL
交错排列的字符串:
7->1->7->2->7->3->7->4->7->7->7->7->NULL
按任意键继续，退出请按'q':
```

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define NULL 0
#define LEN sizeof(struct node)
struct node
```

```

{
char num;
struct node *next;
};
void main()
{
void input(char *p);
struct node *insert(char *p);
void list(struct node *head);
struct node *interlaced(struct node *p1,struct node *p2);
char str1[10],str2[10],c[20],*s1=str1,*s2=str2,*p=c;
struct node *node_a,*node_b,*node_ab;
while(1)
{
printf("请输入字符串 a:\n");
input(s1);
node_a=insert(s1);
printf("字符串链表是:\n");
list(node_a);
printf("请输入字符串 b:\n");
input(s2);
node_b=insert(s2);
printf("字符串链表是:\n");
list(node_b);
node_ab=interlaced(node_a,node_b);
printf("新的交错字符串链表是:\n");
list(node_ab);
printf("\n 按任意键继续, 退出请按'@\n");
if(getchar()=='@')
break;
fflush(stdin);
}
}
void input(char *p)
{
gets(p);
}
struct node *interlaced(struct node *p1,struct node *p2)
{
int length(struct node *head);          //测结点个数
int l1,l2,i;
struct node *p,*p3,*p4;
p3=p1;
p4=p2;
l1=length(p1);
l2=length(p2);
if(l1>=l2)
{
for(i=0;i<l2-1;i++)
{
p3=p3->next;
p4=p4->next;
p1->next=p2;

```

```

        p2->next=p3;
        p1=p3;
        p2=p4;
    }
    p3=p1->next;
    p1->next=p2;
    p2->next=p3;
}
if(l1<l2)
{
    for(i=0;i<l1-1;i++)
    {
        p3=p1->next;
        p4=p2->next;
        p1->next=p2;
        p2->next=p3;
        p1=p3;
        p2=p4;
    }
    p1->next=p2;
}
return(p);
}
struct node *insert(char *p)        //生成节点
{
    struct node *head;
    struct node *p1,*p2;
    int n=0;
    p1=p2=(struct node *)malloc(LEN);
    p1->num=*(p+n);
    head=NULL;
    while(*(p+n)!='\0')        //结束
    {
        n=n+1;
        if(n==1) head=p1;
        else p2->next=p1;
        p2=p1;
        p1=(struct node *)malloc(LEN);
        p1->num=*(p+n);
    }
    p2->next=NULL;
    return(head);
}
void list(struct node *head)        //输出数据
{
    struct node *p;
    p=head;
    if(head!=NULL)
        do
        {
            printf("%c->",p->num);
            p=p->next;
        } while(p!=NULL);
    printf("NULL\n");
}

```

```

    int length(struct node *head)          //测结点个数
    {
        struct node *p;
        int n=1;
        p=head;
        while(p->next!=NULL)
        {
            n++;
            p=p->next;
        }
        return(n);
    }
}

```

```

"E:\10\Debug\10.exe"
请输入字符串a:
123
字符串链表是:
1->2->3->NULL
请输入字符串b:
4567
字符串链表是:
4->5->6->7->NULL
新的交错字符串链表是:
1->4->2->5->3->6->7->NULL
按任意键继续, 退出请按'e'
e
请输入字符串a:
?????
字符串链表是:
?->?->?->?->?->NULL
请输入字符串b:
123
字符串链表是:
1->2->3->NULL
新的交错字符串链表是:
?->1->?->2->?->3->?->?->NULL
按任意键继续, 退出请按'e'
e

```

//第二次做的

```

#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>
#define LEN sizeof(struct letter)
struct letter
{
    char number;
    struct letter *next;
};
void main()
{
    struct letter *insert(char *p);          //生成节点
    void list(struct letter *head);          //输出数据
    struct letter *interlaced(struct letter *p1,struct letter *p2);
}

```



```

char str1[10],str2[10];
struct letter *s1,*s2,*s;
do
{
    fflush(stdin);
    printf("请输入字符串 a\n");
    gets(str1);
    s1=insert(str1);
    printf("请输入字符串 b\n");
    gets(str2);
    s2=insert(str2);
    s=interlaced(s1,s2);
    printf("交错排列的字符串: \n");
    list(s);
    printf("按任意键继续, 退出请按@\n");
}while(getchar()!='@');
}

struct letter *insert(char *p)          //生成节点
{
    struct letter *head;
    struct letter *p1,*p2;
    int n=0;
    p1=p2=(struct letter *)malloc(LEN);
    p1->number=*(p+n);
    head=NULL;
    while(p1->number!='\0')              //结束
    {
        n=n+1;
        if(n==1) head=p1;
        else p2->next=p1;
        p2=p1;
        p1=(struct letter*)malloc(LEN);
        p1->number=*(p+n);
    }
    p2->next=NULL;
    return(head);
}

void list(struct letter *head)          //输出数据
{
    struct letter *p;
    printf("These records are:\n");
    p=head;
    if(head!=NULL)
    do
    {
        printf("%c->",p->number);
        p=p->next;
    } while(p!=NULL);
    printf("NULL\n");
}

struct letter *interlaced(struct letter *head1,struct letter *head2)
{
    int length(struct letter *head);      //测结点个数
    struct letter *p1,*p2,*p3,*p4,*head;
    int i,l1,l2;

```

```

p1=p3=head1;
p2=p4=head2;
l1=length(p1);
l2=length(p2);
head=p1;
if(l1>=l2)
    for(i=0;i<l2;i++)
    {
        p3=p1->next;
        p4=p2->next;
        p1->next=p2;
        p2->next=p3;
        p1=p3;
        p2=p4;
    }
if(l1<l2)
{
    for(i=0;i<l1-1;i++)
    {
        p3=p1->next;
        p4=p2->next;
        p1->next=p2;
        p2->next=p3;
        p1=p3;
        p2=p4;
    }
    p3=p1->next;
    p4=p2->next;
    p1->next=p2;
}
return(head);
}
int length(struct letter *head)    //测结点个数
{
    struct letter *p;
    int n=1;
    p=head;
    while(p->next!=NULL)
    {
        n++;
        p=p->next;
    }
    return(n);
}

```

三、二维矩阵。 一个 6x6 矩阵 a 初始化如下：

```
float a[6][6]={{0.1,0.2,0.3,0.4,0.5,0.6},
               {1.1,1.2,1.3,1.4,1.5,1.6},
               {2.1,2.2,2.3,2.4,2.5,2.6},
               {3.1,3.2,3.3,3.4,3.5,3.6},
               {4.1,4.2,4.3,4.4,4.5,4.6},
               {5.1,5.2,5.3,5.4,5.5,5.6}};
```

请编制一个程序，要求实现如下功能：

1. 函数 **input(i, j)** (10 分)

从键盘输入矩阵 a 的 i 行、j 列参数，若不满足  $0 \leq i \leq 4, 0 \leq j \leq 4$  条件，输出提示信息、重新输入，直到符合条件后，将 i, j 返回给主函数；

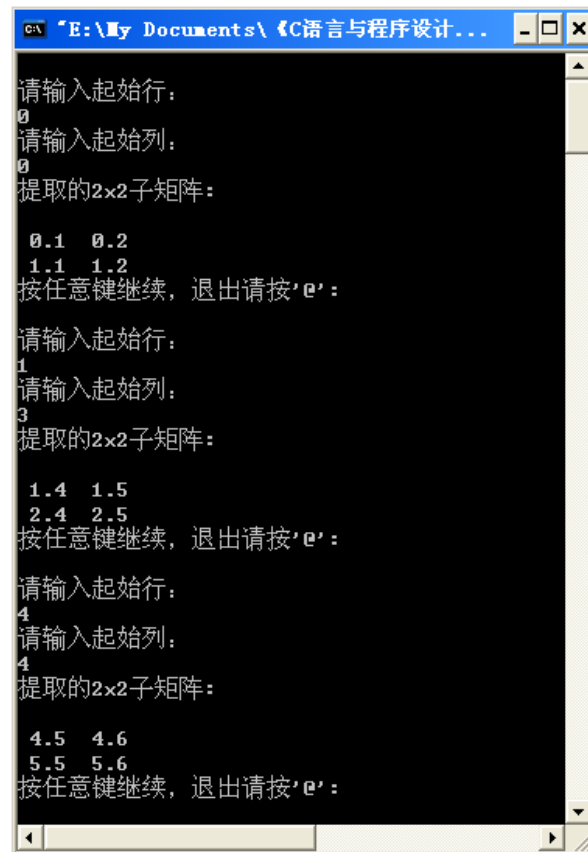
2. 函数 **list(b)** (10 分)

输出子矩阵 b，格式如下图所示。

3. 函数 **submatrix(a, i,j)** (10 分)

对于矩阵 a，以指定元素  $a_{ij}$  为起点，抽取一个 2x2 子矩阵 b，并通过 return 返回给主调函数。

示例：当输入分别为  $a_{00}$ 、 $a_{13}$ 、 $a_{44}$  时，程序输出如下图所示。



```
C:\My Documents\C语言与程序设计...
请输入起始行:
0
请输入起始列:
0
提取的2x2子矩阵:
0.1 0.2
1.1 1.2
按任意键继续, 退出请按'q':

请输入起始行:
1
请输入起始列:
3
提取的2x2子矩阵:
1.4 1.5
2.4 2.5
按任意键继续, 退出请按'q':

请输入起始行:
4
请输入起始列:
4
提取的2x2子矩阵:
4.5 4.6
5.5 5.6
按任意键继续, 退出请按'q':
```

4. 主函数循环运行，分别调用 input()输入 i 和 j、submatrix()抽取子矩阵 b、list()输出 b，当且仅当输入 '@' 时，程序结束运行。(5 分)

#include<stdio.h>

```

#include<stdlib.h>
void main()
{
void input(int *i,int *j);
void list(float *b);
float *submatrix(float a[6][6],int *i,int *j);
    float a[6][6]={0.1,0.2,0.3,0.4,0.5,0.6},
{1.1,1.2,1.3,1.4,1.5,1.6},
{2.1,2.2,2.3,2.4,2.5,2.6},
{3.1,3.2,3.3,3.4,3.5,3.6},
{4.1,4.2,4.3,4.4,4.5,4.6},
{5.1,5.2,5.3,5.4,5.5,5.6}};
float *b;
int i,j;
while(1)
{
    input(&i,&j);
    b=submatrix(a,&i,&j);
    list(b);
    printf("\n 按任意键继续，退出请按'@\n");
    fflush(stdin); //??不加这句不对??
    if(getchar()=='@')
        break;
    fflush(stdin);
}
}
void input(int *i,int *j)
{
while(1)
{
    printf("请输入起始行： \n");
    scanf("%d",i);
    printf("请输入起始列： \n");
    scanf("%d",j);
    if(*i<0||*i>4||*j<0||*j>4)
        printf("输入错误，请重新输入\n");
    else break;
}
}
void list(float *b)
{
int i;
printf("抽取的 2×2 矩阵为\n");
for(i=0;i<4;i++)
{
    printf("%4.1f",*(b+i));
    if((i+1)%2==0)
        printf("\n");
}
}
float *submatrix(float a[6][6],int *i,int *j)
{
float *b;
b=(float *)malloc(sizeof(float)*4);
*(b+0)=*(*(a+*i)+*j);

```

```

*(b+1)=*(*(a+i)+*j+1);
*(b+2)=*(*(a+i+1)+*j);
*(b+3)=*(*(a+i+1)+*j+1);
return(b);
}

```

## 第二次做的

```

#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
void input(int *i,int *j)
{
while(1)
{
printf("请输入行数：");
scanf("%d",i);
printf("请输入列数：");
scanf("%d",j);
if((*i>4||*i<0)||(*j>4||*j<0))
printf("输入错误，请重新输入Wn");
else
break;
}
}
void list(float *b)
{
int i;
printf("子矩阵是:Wn");
for(i=0;i<4;i++)
{
printf("%4.1f",*(b+i));
if((i+1)%2==0)
printf("Wn");
}
}
float *submatrix(float a[][6],int i,int j)
{
float *b;
int k;
b=(float *)malloc(sizeof(float)*4);
for(k=0;k<2;k++)
*(b+k)=*(*(a+i)+j+k);
for(k=0;k<2;k++)
*(b+2+k)=*(*(a+i+1)+j+k);
return(b);
}
void main()
{
void input(int *i,int *j);
float *submatrix(float a[][6],int i,int j);
}

```

```
void list(float *b);  
float a[6][6]={0.1,0.2,0.3,0.4,0.5,0.6},  
{1.1,1.2,1.3,1.4,1.5,1.6},  
{2.1,2.2,2.3,2.4,2.5,2.6},  
{3.1,3.2,3.3,3.4,3.5,3.6},  
{4.1,4.2,4.3,4.4,4.5,4.6},  
{5.1,5.2,5.3,5.4,5.5,5.6}};  
int i,j;  
float *b;  
do  
{  
    input(&i,&j);  
    b=submatrix(a,i,j);  
    list(b);  
    printf("按任意键继续, 退出请按@Wn");  
    fflush(stdin);  
} while(getchar()!='@');  
}
```

四、**二维矩阵**。矩阵 a 及程序功能如同第三题，但从 a 中提取的子矩阵 b 的大小，是根据键盘输入动态指定的。其中：

1. **函数 DynamicInput(oldrl,oldcl,newrl,newcl) (2 分)**

在主函数中调用 **DynamicInput()**,从键盘读取下列参数，并返回给主函数

oldrl: 矩阵 a 起始点的行号;

oldcl: 矩阵 a 起始点的列号;

newrl: 子矩阵 b 的行数;

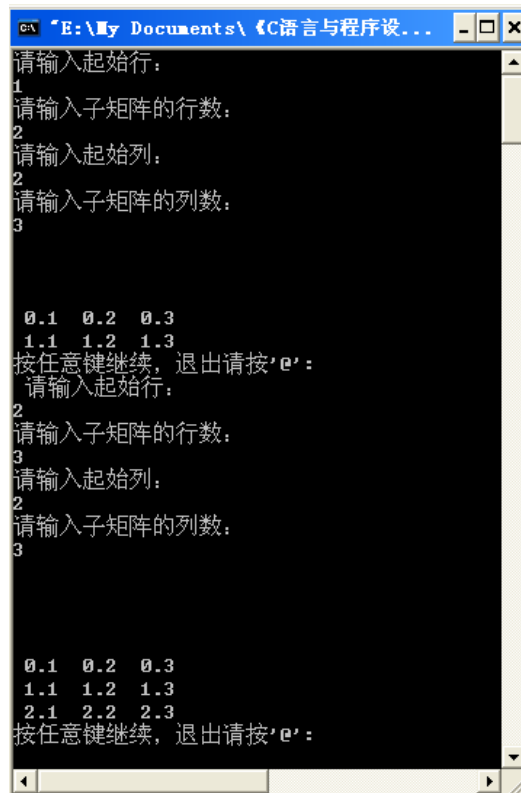
newcl: 子矩阵 b 的列数。

2. **函数 DynamicSsubmatrix(a, oldrl,oldcl,newrl,newcl) (2 分)**

在主函数中调用 **DynamicSsubmatrix()**,对给定的 6x6 矩阵 a，根据参数 **oldrl,oldcl,newrl,newcl**，以 **aoldrl,oldcl** 为起点，动态的提取一个 **newrl** 行、**newcl** 列的子矩阵 b，并通过 return 返回给主函数。

3. **函数 DynamicList() (1 分)**

在主函数中调用 **DynamicList ()** 输出子矩阵 b。示例如下：



```
CA "E:\My Documents\《C语言与程序设... - 窗口 x
请输入起始行:
1
请输入子矩阵的行数:
2
请输入起始列:
2
请输入子矩阵的列数:
3

0.1 0.2 0.3
1.1 1.2 1.3
按任意键继续，退出请按'q':
请输入起始行:
2
请输入子矩阵的行数:
3
请输入起始列:
2
请输入子矩阵的列数:
3

0.1 0.2 0.3
1.1 1.2 1.3
2.1 2.2 2.3
按任意键继续，退出请按'q':
```

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    void DynamicInput(int *oldrl,int *oldcl,int *newrl,int *newcl);
    float *DynamicSsubmatrix(float a[6][6],int *oldrl,int *oldcl,int *newrl,int *newcl);
    void DynamicList(float *b,int *oldcl,int *newcl);
    int or,oc,nr,nc;
    float a[6][6]={{0.1,0.2,0.3,0.4,0.5,0.6},
```

```

{1.1,1.2,1.3,1.4,1.5,1.6},
{2.1,2.2,2.3,2.4,2.5,2.6},
{3.1,3.2,3.3,3.4,3.5,3.6},
{4.1,4.2,4.3,4.4,4.5,4.6},
{5.1,5.2,5.3,5.4,5.5,5.6}};
float *b;
while(1)
{
    DynamicInput(&or,&oc,&nr,&nc);
    b=DynamicSubmatrix(a,&or,&oc,&nr,&nc);
    DynamicList(b,&oc,&nc);
    printf("\n 按任意键继续，退出请按'@\n");
    fflush(stdin); //??不加这句不对??
    if(getchar()=='@')
        break;
    fflush(stdin);
}
}
void DynamicInput(int *oldrl,int *oldcl,int *newrl,int *newcl)
{
    printf("请输入起始行: \n");
    scanf("%d",oldrl);
    printf("请输入行数: \n");
    scanf("%d",oldcl);
    printf("请输入起始列: \n");
    scanf("%d",newrl);
    printf("请输入列数: \n");
    scanf("%d",newcl);
}
float *DynamicSubmatrix(float a[6][6],int *oldrl,int *oldcl,int *newrl,int *newcl)
{
    int i,j;
    float *b;
    b=(float *)malloc(sizeof(float)*(*oldcl)*(*newcl));
    for(i=0;i<(*oldcl);i++)
        for(j=0;j<(*newcl);j++)
        {
            *(b+i*(*newcl)+j)=a[*oldrl+i-1][*newrl+j-1];
        }
    return(b);
}
void DynamicList(float *b,int *oldcl,int *newcl)
{
    int i,j;
    printf("抽取的%d×%d 矩阵为\n",*oldcl,*newcl);
    for(i=0;i<(*oldcl);i++)
    {
        for(j=0;j<(*newcl);j++)
            printf("%4.1f",*(b+i*(*oldcl)+j));
        printf("\n");
    }
}

```



