**实验06　指针**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 班级 |  | 学号 |  | 姓名 |  |

**一、实验目的**

1．掌握

1. 掌握指针和间接访问的概念，会定义和使用指针变量。
2. 能正确使用数组的指针和指向数组的指针变量。
3. 能正确使用字符串的指针和指向字符串的指针变量。

**二、实验要求**

1. 请在实验课前将本实验要求编写的程序打好草稿；
2. 实验后所有学生要提交电子版实验报告和源程序；

**三、实验内容和步骤**

编程序并上机调试运行以下程序(**都要求用指针处理**)。

1. 输入3个整数，按由小到大的顺序输出，然后将程序改为：输入3个字符串，按由小到大顺序输出。
2. 先编写一个程序，在主函数中输入**3**个整数，在一个自定义的函数中交换两个整数，在主函数中按由小到大的顺序输出这三个整数。  
   在调用和定义交换函数时要求：要交换的两个整数的指针作为实参；整型指针变量作为形参。

**实验文档整理方法**：

① 将编写的源程序存为**6-01.C**；

② 将源程序的内容复制并粘贴到下面的表格中。

|  |  |
| --- | --- |
| 程序 | #include<stdio.h>  int main (void)  {  int a,b,c;  void swap(int \*x,int \*y);  printf("请输入三个整数：");  scanf("%d%d%d",&a,&b,&c);  if(a>b)  swap(&a,&b);  if(a>c)  swap(&a,&c);  if(b>c)  swap(&b,&c);  printf("%2d%2d%2d",a,b,c);  return 0;  }  void swap(int \*x,int \*y)  {  int temp;  temp=\*x;  \*x=\*y;  \*y=temp;  }  1 |

1. 把程序改为能处理输入3个字符串，按由小到大的顺序输出。

**实验文档整理方法**：

① 将编写的源程序存为**6-02.C**；

② 将源程序的内容复制并粘贴到下面的表格中。

|  |  |
| --- | --- |
| 程序 | #include<stdio.h>  #include<string.h>  int main (void)  {  char str1[10],str2[10],str3[10];  void swap(char \*x,char \*y);  printf("请输入三个字符：");  scanf("%s%s%s",str1,str2,str3);  if(strcmp(str1,str2)>0)  swap(str1,str2);  if(strcmp(str1,str3)>0)  swap(str1,str3);  if(strcmp(str2,str3)>0)  swap(str2,str3);  printf("%s%s%s",str1,str2,str3);  return 0;  }  void swap(char \*x,char \*y)  {  char temp;  temp=\*x;  \*x=\*y;  \*y=temp;  }  2 |

1. 编写一个用户自定义函数Sort，使用简单选择法对具有10个整型元素的一维数组排序（升序），要求：
2. 实参：一维数组名，形参：一维数组。

**实验文档整理方法**：

① 将编写的源程序存为**6-03.C**；

② 将源程序的内容复制并粘贴到下面的表格中。

|  |  |
| --- | --- |
| 程序 | #include<stdio.h>  int main(void)  {  int a[10];  int i;  void Sort(int m[10]);  printf("请输入10个数：");  for(i=0;i<10;i++)  scanf("%d",&a[i]);  Sort(a);  return 0;  }  void Sort(int m[10])  {  int i,j,k,temp;  for(i=0;i<9;i++)  {  k=i;  for(j=i+1;j<10;j++)  {  if(m[k]>m[j])  {  k=j;  temp=m[i];  m[i]=m[k];  m[k]=temp;  }  }  }  printf("排序结果为：");  for(i=0;i<10;i++)  printf("%2d",m[i]);    }  3 |

1. 实参：一维数组名，形参：指向一维数组元素的指针变量。

**实验文档整理方法**：

① 将编写的源程序存为**6-04.C**；

② 将源程序的内容复制并粘贴到下面的表格中。

|  |  |
| --- | --- |
| 程序 | #include<stdio.h>  int main(void)  {  int a[10];  int i;  void Sort(int m[10]);  printf("请输入10个数：");  for(i=0;i<10;i++)  scanf("%d",&a[i]);  Sort(a);  return 0;  }  void Sort(int \*m)  {  int i,j,k,temp;  for(i=0;i<9;i++)  {  k=i;  for(j=i+1;j<10;j++)  {  if(m[k]>m[j])  {  k=j;  temp=m[i];  m[i]=m[k];  m[k]=temp;  }  }  }  printf("排序结果为：");  for(i=0;i<10;i++)  printf("%2d",m[i]);    }  4 |

1. 实参：指向一维数组元素的指针变量，形参：一维数组。

**实验文档整理方法**：

① 将编写的源程序存为**6-05.C**；

② 将源程序的内容复制并粘贴到下面的表格中。

|  |  |
| --- | --- |
| 程序 | #include<stdio.h>  int main(void)  {  int a[10];  int i;  int \*pa=a;  void Sort(int m[10]);  printf("请输入10个数：");  for(i=0;i<10;i++)  scanf("%d",&a[i]);  Sort(pa);  return 0;  }  void Sort(int m[10])  {  int i,j,k,temp;  for(i=0;i<9;i++)  {  k=i;  for(j=i+1;j<10;j++)  {  if(m[k]>m[j])  {  k=j;  temp=m[i];  m[i]=m[k];  m[k]=temp;  }  }  }  printf("排序结果为：");  for(i=0;i<10;i++)  printf("%2d",m[i]);    }  5 |

1. 实参：指向一维数组元素的指针变量，形参：指向一维数组元素的指针变量。

**实验文档整理方法**：

① 将编写的源程序存为**6-06.C**；

② 将源程序的内容复制并粘贴到下面的表格中。

|  |  |
| --- | --- |
| 程序 | #include<stdio.h>  int main(void)  {  int a[10];  int i;  int \*pa=a;  void Sort(int m[10]);  printf("请输入10个数：");  for(i=0;i<10;i++)  scanf("%d",&a[i]);  Sort(pa);  return 0;  }  void Sort(int \*m)  {  int i,j,k,temp;  for(i=0;i<9;i++)  {  k=i;  for(j=i+1;j<10;j++)  {  if(m[k]>m[j])  {  k=j;  temp=m[i];  m[i]=m[k];  m[k]=temp;  }  }  }  printf("排序结果为：");  for(i=0;i<10;i++)  printf("%2d",m[i]);    }  5 |

1. 编写一个函数，求出一个3行4列的整型二维数组周边元素之和。  
   在主函数中用scanf函数输入二维数组的各个元素初值。
2. 将数组0行0列元素的地址作为函数实参。

**实验文档整理方法**：

① 将编写的源程序存为**6-07.C**；

② 将源程序的内容复制并粘贴到下面的表格中。

|  |  |
| --- | --- |
| 程序 | #include<stdio.h>  int main(void)  {  int i,j;  int a[3][4];  void sum (int \*p);  printf("请输入数值：");  for(i=0;i<3;i++)  for(j=0;j<4;j++)  scanf("%d",&a[i][j]);  printf("数组为：\n");  for(i=0;i<3;i++)  {  for(j=0;j<4;j++)  printf("%4d",a[i][j]);  printf("\n");  }  sum(\*a);  return 0;    }  void sum (int \*p)  {  int m[3][4];  int a,b;  int sum=0;  p=&m[0][0];  for(a=0;a<3;a++)  sum= m[a][0]+m[a][3]+sum;  for(b=1;b<3;b++)  sum=m[0][b]+m[2][b]+sum;  printf("周边元素之和为：%d",sum);  } |

1. 将数组0行的地址作为函数实参。

**实验文档整理方法**：

① 将编写的源程序存为**6-08.C**；

② 将源程序的内容复制并粘贴到下面的表格中。

|  |  |
| --- | --- |
| 程序 | #include<stdio.h>  int main(void)  {  int i,j;  int a[3][4];  void sum (int (\*p)[4]);  printf("请输入数值：");  for(i=0;i<3;i++)  for(j=0;j<4;j++)  scanf("%d",&a[i][j]);  printf("数组为：\n");  for(i=0;i<3;i++)  {  for(j=0;j<4;j++)  printf("%4d",a[i][j]);  printf("\n");  }  sum((\*a)[4]);  return 0;    }  void sum (int (\*p)[4])  {  int m[3][4];  int a,b,i;  int sum=0;  for(i=0;i<4;i++)  p=m;  for(a=0;a<3;a++)  sum= m[a][0]+m[a][3]+sum;  for(b=1;b<3;b++)  sum=m[0][b]+m[2][b]+sum;  printf("周边元素之和为：%d",sum);  } |

**四、实验报告提交方法**

将实验过程中生成的**C**源程序文件**6-01.C**、**6-02.C**、**6-03.C、6-04.C、6-05.C**、**6-06.C**、**6-07.C、6-08.C**与本文档压缩在一个名为你的**学号.rar**的文件中（假如你的学号是**123**，那么压缩文件的名字就为**123.rar**），并作为附件提交即可。