**《C 语言程序设计》**

**（第二版）**

**教案**

**北京出版社**

**课时分配表**

|  |  |  |  |
| --- | --- | --- | --- |
| **章序** | **课程内容** | **课时** | **备注** |
| **1** | **认识 C 程序** | **6** |  |
| **2** | **C 程序设计基础** | **7** |  |
| **3** | **顺序结构程序设计** | **5** |  |
| **4** | **选择结构程序设计** | **3** |  |
| **5** | **循环结构程序设计** | **6** |  |
| **6** | **数组** | **5** |  |
| **7** | **用户自定义函数** | **7** |  |
| **8** | **编译预处理** | **5** |  |
| **9** | **指针** | **9** |  |
| **10** | **构造数据类型** | **5** |  |
| **11** | **文件读写技术** | **6** |  |
| **12** | **C 语言综合实训** | **4** |  |
| **13** | **附录** | **4** |  |
| **总计** |  | **72** |  |

**第9课 指针**

|  |  |  |
| --- | --- | --- |
| **课 题** | 指针 | |
| **课 时** | 9课时（405 min）。 | |
| **教学目标** | **知识技能目标：**  1．认识指针。  2．通过学习与练习掌握指针访问基本变量。  **思政育人目标：**  让学生通过学习指针，能与人积极交流沟通，具备乐观向上的态度，具备开发高效程序的思维能力。 | |
| **教学重难点** | **教学重点：**认识指针  **教学难点：**指针访问基本变量 | |
| **教学方法** | 讲授法、问答法、讨论法 | |
| **教学用具** | 电脑、投影仪、多媒体课件、教材 | |
| **教学设计** | 第1节课：考勤（2min）--知识讲解（40min）--作业布置（3min）  第2节课：知识讲解（40min）--课堂小结（3min）--作业布置（2min）  第3节课：知识讲解（40min）--课堂小结（3min）--作业布置（2min）  第4节课：知识讲解（40min）--课堂小结（3min）--作业布置（2min）  第5节课：知识讲解（40min）--课堂小结（3min）--作业布置（2min）  第6节课：知识讲解（40min）--课堂小结（3min）--作业布置（2min）  第7节课：知识讲解（40min）--课堂小结（3min）--作业布置（2min）  第8节课：知识讲解（40min）--课堂小结（3min）--作业布置（2min）  第9节课：知识讲解（40min）--课堂小结（3min）--作业布置（2min） | |
| **教学过程** | **主 要 教 学 内 容 及 步 骤** | **设计意图** |
| **考勤**  **（2min）** | ■【教师】清点上课人数，记录好考勤  ■【学生】班干部报请假人员及原因 | 培养学生的组织纪律性,掌握学生的出勤情况 |
| **知识讲解**  （40min） | **【教师】**认识指针  **任务描述**  有这样一个故事：  革命战争时期，地下工作者阿金接到上级指令，要去寻找打开加密电码的密钥（这是一个整数）。几经周折，才探知如下线索：密钥藏在一栋三年前就被贴上封条的小楼中。一个风雨交加的夜晚，阿金潜入了小楼。房间很多，不知该进哪一间。正在一筹莫展之际，忽然走廊上的电话铃响起。阿金毫不迟疑，抓起话筒，只听一个陌生人说：“去打开 211 房间，那里有线索。”阿金疾步上楼打开 211 房间，用电筒一照，只见桌上赫然 6 个大字：地址 1000。阿金眼前一亮，迅速找到 1000 房间，取出重要数据 66，完成了任务。  结合这个故事，理解指针、指针变量等相关概念。  **预备知识**  在计算机中，所有的数据都是存放在存储器中的。为了正确地访问这些存储单元，必须为每个存储单元编上号。根据一个内存单元的编号即可找到该内存单元。内存单元的编号也称为“地址”，在程序设计语言中称为“指针”。与指针相关的概念示意图如图 9-1 所示。    **任务实施**  根据图9-1 可知：  1. 要寻找的密钥（整数）是66。它藏在一个地址单元中，地址是1000。  2. 地址1000 又由p 单元所指认，p 单元的地址是211。  3. 66 的直接地址是1000；66 的间接地址是211；211 中存的是直接地址1000。  4. a 是整型变量，存放数据66；p 是指针变量，内装地址1000。  结论：  1. 指针：内存地址。  2. 指针变量：存储内存地址的变量。  3. 把地址赋给指针变量p，即称为该变量指向该地址。  **【学生】**思考、讨论。 | **展示认识指针，让学生更加仔细的阅读，从而激发学生的学习欲望。** |
| **作业布置**（3min） | **【教师】**布置课后作业  简述指针、指针变量的概念。 | 通过课后练习，使学生巩固所学新知识 |
| **知识讲解**  （40min） | **【教师】**指针访问基本变量  **任务描述**  输入 a、b 两个整数，利用指针，将它们按从大到小的顺序输出。  **预备知识**  **一、指针变量的定义**  C 语言规定，变量在使用之前必须先定义。和整型变量、实型变量一样，指针变量在使用之前也要先定义。  指针变量的定义形式如下：  类型标识符 \* 指针变量名；其中，“\*”是一个说明符，用来说明其后的变量是一个指针变量。“类型标识符”用来说明该指针变量存放哪一种类型变量的地址，它可以是任何一个合法的 C 语言数据类型。  例如：  　　int a=12,\*p;  　　p=&a;  该语句定义了一个 int 类型的简单变量 a 和一个 int 类型的指针变量 p，并且指针变量 p 被初始化为整型变量 a 的地址（即 p 指向 a），如图 9-2 所示。    【注意】  （1）定义指针变量时的“\*”表示该变量为指针变量，但指针变量名为 p，而不是 \*p。  （2）一个指针变量只能指向同一个类型的变量，例如，p 不能忽而指向一个整型变量，忽而指向一个实型变量，因此必须规定指针变量所指向的变量的类型。  **二、指针变量的初始化**  指针变量在定义的同时被赋以初值，叫做指针变量的初始化。指针变量初始化的一般格式为：  类型标识符 \* 指针变量名 = 初始地址值 ;  例如：  　　int x;  　　int \*p1=&x;  【注意】  （1）这里的初始化是对指针变量的初始化，而不是对指针所指变量的初始化。例如，上述例子中是把整型变量 x 的地址赋给了指针变量 p1，而变量 x 还没有初值。  （2）指针所指向对象的数据类型必须与指针的数据类型一致。例如：  　　fl oat x;  　　int \*p1=&x;  是错误的。  （3）可以把一个指针的值赋给另一个指针。  例如：  　　int x;  　　int \*q2,\*q1=&x;  　　q2=q1;  （4）可以把一个指针初始化为一个空指针。  例如：  　　int \*q=NULL;  （5）当把一个变量的地址作为初值赋给指针变量时，这个变量必须在指针初始化之前已经定义过，因为没定义过的变量没有地址。  例如：  fl oat f;  fl oat \*w=&f; /\* 该定义语句之前 ,f 必须已经被定义 \*/  **三、地址与指针运算符**  1. 地址运算符“&”  例如，假如 p 是指针变量，a 为整型变量，则：  　　p=&a;  此条语句中“&”的功能是：把变量 a 的地址取出来赋给指针变量 p。  2. 指针运算符“\*”  例如，假如 p 仍为指针变量，a、b 为整型变量，则：  　　p=&a;  　　b=\*p; /\* \*p 是 p 所指变量 , 等价于 a \*/  【注意】p，\*p 和 &p 三者的区别。  　p：是指针变量，其内容是某一变量的地址。  　\*p：是指针变量所指向的变量，其内容是 p 所指变量的值。  　&p：是指针变量本身所占据的内存地址。  另外，“++”“--”“\*”和“&”的优先级相同，而且具备右结合性。  **【学生】**思考、讨论。 | **通过教师讲解，了解指针访问基本变量的基本理论知识。** |
| **课堂小结**  （3min） | 【**教师**】**回顾和总结本节课的知识点。**  **这节课我们一起学习了指针访问基本变量，让学生能与人积极交流沟通，具备乐观向上的态度，具备开发高效程序的思维能力。** | 通过对所学知识的回顾，培养学生的归纳总结能力 |
| **作业布置**（2min） | **【教师】**布置课后作业  阅读下列程序段，回答相应的问题。  int a=3,b=7,\*p1,\*p2;  p1=&a;  p2=&b;  p1=p2; /\* 该语句的含义是？ \*/  \*p1=\*p2; /\* 该语句的含义是？ \*/ | 通过课后练习，使学生巩固所学新知识 |
| **知识讲解**（45min） | **【教师】**指针与一维数组  **任务描述**  键盘输入 10 个整数，利用指针引用数组元素的形式输出数组各元素的值。  **预备知识**  **一、指向数组的指针变量和数组元素的地址**  C 语言规定，数组名代表数组在内存中的起始地址。如果定义了一个指针变量，并将数组的首地址赋给该指针变量，则该指针就指向了这个一维数组。  例如：  fl oat a[3],\*p1,\*p2;  p1=&a[0];  p2=a;  其中，a 是数组名，代表数组的首地址，&a[0] 是数组元素 a[0] 的地址。由于 a[0]的地址就是数组的首地址，所以，两条赋值操作效果完全相同，指针变量 p1 和 p2 都指向数组 a。  假如正确执行了：  int a[10],\*p;  p=a;  **二、通过指针引用数组元素**  引入指针变量后，就可以用两种方法访问数组元素。  （1）下标法。即用 a[i] 的形式访问数组元素，在单元六中介绍数组时都是采用这种方法来访问数组元素的。  （2）指针法。即采用 \*（p+i）或 \*（a+i）的形式，用间接访问的方法来访问数组元素。  【注意】a 是数组名，虽然看起来像一个指针变量，但实质上它是常量。它的值在程序运行期间是不变的，所以假如出现“a++；”或“a=a+i；”，这样的语句是不合法的。  仍然有：  int a[10],\*p;  p=a;  **任务实施**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* 程序: program9\_3.c \* /  /\* 功能: 键盘输入10 个数, 利用指针引用数组元素的形式输出数组各元素的值。 \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  void main()  {  int i,a[10],\*ptr=a;  printf(" 请输入10 个整数:");  for(i=0;i<=9;i++)  scanf("%d",ptr++);  printf(" 输出10 个整数:");  ptr=a;  for(i=0;i<=9;i++)  printf("%4d",\*(ptr++));  }  程序运行结果：  请输入10 个整数:1 2 3 4 5 6 7 8 9 0 ↙  输出10 个整数:1 2 3 4 5 6 7 8 9 0  **【学生】**思考、讨论。 | **教师通过指针与一维数组展示，让学生了解指针与一维数组的基本理论知识。** |
| **课堂小结**  （3min） | 【**教师**】**回顾和总结本节课的知识点。**  **这节课我们一起学习了指针与一维数组，让学生能清楚认识指针访问数组越界造成的严重后果，养成细心、周密的软件服务意识。** | 通过对所学知识的回顾，培养学生的归纳总结能力 |
| **作业布置**（2min） | 【**教师**】**布置课后作业**  **用键盘输入 10 个数，用指针引用数组元素的形式，逆序输出数组各元素的值。** | 通过课后练习，使学生巩固所学新知识 |
| **知识讲解**（45min） | **【教师】**指针与二维数组  **任务描述**  利用指向二维数组的指针，输入输出二维数组的值。  **预备知识**  **一、二维数组的地址表示方法**  定义一个二维数组：  int a[3][4];  表示二维数组有 3 行 4 列共 12 个元素，也可以把 a 看成一个一维数组，它有 3 个元素：a[0]、a[1]、a[2]。二维数组的元素在内存中是按行存放的。  其中 a 是二维数组的首地址，&a[0][0] 既可以看作数组 0 行 0 列的首地址，也可以看作是二维数组的首地址。a[0] 是第 0 行的首地址，当然也是数组的首地址。同理 a[n]就是第 n 行的首址；&a[n][m] 就是元素 a[n][m] 的地址。  设数组 a 的首地址为 1000，可以推出以下几点：  （1）a[0] 是第一个一维数组的数组名和首地址，因此也为 1000。  （2）\*（a+0）或 \*a 是与 a[0] 等效的，它表示一维数组 a[0] 的 0 号元素的首地址，也为 1000。  （3）&a[0][0] 是二维数组 a 的 0 行 0 列元素的首地址，同样是 1000。因此，a、a[0]、\*（a+0）、\*a 和 &a[0][0] 是相等的。  按同样的方法分析可得：  （1）a+1 是二维数组 1 行的首地址，等于 1008。  （2）a[1] 是第二个一维数组的数组名和首地址，也为 1008。  （3）&a[1][0] 是二维数组 a 的 1 行 0 列元素首地址，也是 1008。  因此，a+1、a[1]、\*（a+1）、\*a 和 &a[1][0] 是相等的。  由此可得，a+i、a[i]、\*（a+i）、&a[i][0] 是等同的。  此外，&a[i] 和 a[i] 也是等同的。因为在二维数组中不能把 &a[i] 理解为元素 a[i] 的地址，因为不存在元素 a[i]。  然后具体到一个数组元素，如 a[0]，可以看成是 a[0]+0，它是一维数组 a[0] 的 0 号元素的首地址，而 a[0]+1 则是一维数组 a[0] 的 1 号元素的首地址，由此可得 a[i]+j 则是一维数组 a[i] 的 j 号元素的首地址，它等于 &a[i][j]。由 a[i]= \*（a+i）得 a[i]+j=\*（a+i）+j，由于 \*（a+i）+j 是二维数组 a 的 i 行 j 列元素的首地址，该元素的值等于 \*（\*（a+i）+j）。  可见，二维数组地址表示的形式很多，每种地址形式也都有对应的数组元素引用方法。总结如下：  二维数组元素地址：① &a[i][j] 　　② a[i]+j 　③ \*（a+i）+j  二维数组元素引用：① a[i][j] 　　② \*（a[i]+j） 　③ \*（\*（a+i）+j）  二、二维数组的指针变量  二维数组指针变量说明的一般形式为：  类型说明符 (\* 指针变量名 )[ 长度 ];  【说明】  （1）“类型说明符”为所指数组的数据类型。  （2）“\*”表示其后的变量是指针类型。  （3）“长度”表示二维数组分解为多个一维数组时的一维数组的长度，即二维数组的列数。  （4）（\* 指针变量名）两边的括号不可少，如缺少括号则表示是指针数组（详见单元九任务 6），意义就完全不同了。  例如：把二维数组 a[3][4] 分解为一维数组 a[0]、a[1]、a[2] 之后，设 p 为指向二维数组的指针变量。可定义为：  int (\*p)[4];  它表示 p 是一个指针变量，指向二维数组 a 或指向第一个一维数组 a[0]，其值等于 a，a[0] 或 &a[0][0] 等。而 p+i 则指向一维数组 a[i]。  由此得出 \*（p+i）+j 是二维数组 i 行 j列元素的地址，而 \*（\*（p+i）+j）则是 i 行 j 列元素的值。  **【学生】**思考、讨论。 | **教师通过指针与二维数组展示，让学生了解指针与二维数组的基本理论知识。** |
| **课堂小结**  （3min） | 【**教师**】**回顾和总结本节课的知识点。**  **这节课我们一起学习了指针与二维数组，让学生具备良好的代码规范，能提升代码质量，减少低级的代码错误。** | 通过对所学知识的回顾，培养学生的归纳总结能力 |
| **作业布置**（2min） | 【**教师**】**布置课后作业**  **该程序将二维数组的 12 个元素输出在同一行上，若想输出 3 行 4 列的形式，应该如何修改程序的输出部分？** | 通过课后练习，使学生巩固所学新知识 |
| **知识讲解**（45min） | **【教师】**指针与字符串  **任务描述**  查找输入的字符串中有无字母“a”。  **任务分析**  1. 用指针指向一个字符数组；  2. 输入一个字符串存入该数组中；  3. 让指针变量从字符串首部开始依次向字符串尾部移动，每移动一次，将指针所指向的字符与字符“a”比较，若相等，则显示有字符“a”，可立刻结束判断；若不相等，继续比较；  4. 假如遇到了字符串结束标志“\0”还未遇见“a”，则显示没有字符“a”。  **预备知识**  **一、字符指针与字符串指针**  字符串指针变量的定义与字符指针变量的定义是相同的，所以只能按对指针变量的赋值的不同来区别。对指向字符变量的指针变量应赋予该字符变量的地址。例如：  char c,\*p1,\*p2;  p2=&c; /\* 把字符变量 c 的地址赋给 p2 \*/  p1="C Language"; /\* 把字符串 C Language 的首地址赋给 p1 \*/  \*p2='C'; /\* 等价于 c='C' \*/  **二、字符串指针与字符数组**  C 语言中，可以用两种方法实现一个字符串，即字符数组和字符串指针。  在单元六中，我们已经详细学习过字符数组的知识。字符数组不仅可以用来存放字符，也可以用来存放字符串。字符串指针变量也可以实现字符串的存储和运算，但两者是有区别的。在具体使用时，要注意以下几个问题：  （1）本质上的区别。字符串指针变量本身是一个变量，用于存放字符串的首地址；字符数组是由若干个数组元素组成的，可以用来存放整个字符串。  （2）变量的初始化上有区别。对字符数组作初始化赋值，必须采用外部类型或静态类型，例如：  static char st[]={"C Language"};  而对字符串指针变量则无此限制，如：  char \*p="C Language";  （3）二者赋值方式上有区别。对字符串指针方式，例如：  char \*p="C Language";  可写成  char \*p;  p="C Language";  而对数组方式：  static char st[]={"C Language"};  不能写成：  char st[20];  st={"C Language"};  而只能对字符数组的各个元素逐个赋值。  **任务实施**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* 程序 : program9\_6.c \*/  /\* 功能 : 查找输入的字符串中有无字母 "a"。 \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  void main()  {  char st[20],\*p;  int i;  printf(" 请输入一个字符串 :");  p=st;  scanf("%s",p);  for(i=0;\*(p+i)!='\0';i++)  {  if(\*(p+i)== 'a')  { printf(" 该字符串中有字符 'a'!\n");  break;  }  }  if(\*(p+i)=='\0') printf(" 该字符串中没有字符 'a'!\n");  }  程序运行结果：  请输入一个字符串 :This is a C program! ↙  该字符串中有字符‘a’！  **【学生】**思考、讨论。 | **教师通过指针与字符串展示，让学生了解指针与字符串的基本理论知识。** |
| **课堂小结**  （3min） | 【**教师**】**回顾和总结本节课的知识点。**  **这节课我们一起学习了指针与字符串，让学生能够建立自己的知识体系，具有一定层面的编程思想，具备不断跟进新技术的能力。** | 通过对所学知识的回顾，培养学生的归纳总结能力 |
| **作业布置**（2min） | 【**教师**】**布置课后作业**  **写出下列程序的运行结果。**  **#include <stdio.h>**  **#include <string.h>**  **void main()**  **{**  **char c,\*p1,\*p2;**  **p2=&c;**  **p1="C Language";**  **\*p2='C';**  **printf("%s,%c\n",p1,\*p2);**  **}** | 通过课后练习，使学生巩固所学新知识 |
| **知识讲解**（45min） | **【教师】**指针数组  **任务描述**  设计一个程序，将 3 个字符串常量按字母顺序输出。  **任务分析**  1. 定义一个指针数组，内赋三个字符串的地址；  2. 借助 strcmp 函数对指针数组排序；  3. 输出指针数组。  **预备知识**  数组元素类型均为指针类型的数组，称为指针数组。也就是说，指针数组中的每一个元素都是指针变量。指针数组比较适合用来指向若干个字符串，使字符串处理更加方便灵活。  **一、指针数组的定义**  指针数组的定义格式如下：  类型标识符 \* 数组名 [ 常量表达式 ];  例如：  int \*p[10];  定义了一个指针数组 p[10]，它有 10 个元素。这 10 个元素均为指针变量，每个数组元素（指针变量）都可指向一个整型变量。  【注意】指针数组 int \*p[4] 与指向数组的指针 int (\*p)[4] 是两个完全不同的概念，两者在定义格式上十分相似，仅差一对小括号，但前者是数组，后者是指针，具体区别如下。  （1）int \*p[4]；  由于中括号比“\*”号优先级高，因此 p 先与 [4] 结合，形成 p[4] 形式，这表明 p 是一维数组，它有 4 个元素。p 前面的“int \*”表明 p 是指针数组，每个元素都是一个指向整型变量的指针变量。  （2）int （\*p）[4]；  由于“( )”比“[]”号优先级高，因此 p 先与“\*”结合，这表明 p 是指针变量，p后面的 [4]，表明指针变量 p 指向一个一维数组，这个数组有 4 个元素。最前面的 int 决定数组元素的类型为整型。  **二、指针数组的初始化**  与其他类型的数组一样，指针数组在定义的同时也可以被初始化。初始化后的指针数组常用来保存指向字符串的指针。  例如：  char \*str1[]={"structure","language","time"};  str1 是一个指针数组，它在定义的同时被初始化。由初值表中的初值个数可以看出，指针数组 str1 有 3 个元素，每个元素都是一个字符指针，存放的是字符串的首地址。其中 str1[0] 指向字符串“structure”，str1[1] 指向字符串“language”，str1[2] 指向字符串“time”。  再例如：  char str2[3][10]={ "structure","language","time"};  str2 是二维字符数组，它可以“存放”3 个字符串，每个字符串的最大长度为 9。  **任务实施**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* 程序 : program9\_7.c \*/  /\* 功能 : 设计一个程序 , 将 3 个字符串常量按字母顺序输出。 \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  #include <string.h>  void main()  {  char \*lesson[3]={ "data structure", "computer design", "c  language"};  int i;  char \*cp;  if(strcmp(lesson[0], lesson[1])>0)  { cp=lesson[0];lesson[0]=lesson[1];lesson[1]=cp;}  if(strcmp(lesson[0], lesson[2])>0)  { cp=lesson[0];lesson[0]=lesson[2];lesson[2]=cp;}  if(strcmp(lesson[1], lesson[2])>0)  { cp=lesson[1];lesson[1]=lesson[2];lesson[2]=cp;}  for(i=0;i<3;i++)  printf("\n%s", lesson[i]);  }  程序运行结果：  c language  computer design  data structure  lesson 数组中各元素的初始状态与终止状态。  **【学生】**思考、讨论。 | **教师通过指针数组展示，让学生了解指针数组的基本理论知识。** |
| **课堂小结**  （3min） | 【**教师**】**回顾和总结本节课的知识点。**  **这节课我们一起学习了指针数组，让学生能弄清指针数组的内容与所指的关系，关注核心价值。** | 通过对所学知识的回顾，培养学生的归纳总结能力 |
| **作业布置**（2min） | 【**教师**】**布置课后作业**  **简述指针数组的初始化。** | 通过课后练习，使学生巩固所学新知识 |
| **知识讲解**（45min） | **【教师】**指针做函数参数  **任务描述**  利用指针做函数参数，实现 a、b 两个变量值的互换。  **任务分析**  1. main( ) 函数调用 swap 函数时，将变量 a、b 的地址传递给对应的形参 pt1 和 pt2；  2. 互换 pt1 和 pt2 所指变量的值，实质就是互换 a 和 b 的值。  **任务实施**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* 程序 : program9\_8.c \*/  /\* 功能 : 利用指针做函数参数 , 实现变量 a,b 值的互换。 \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  void main()  {  void change();  int \*p1,\*p2,a,b;  printf(" 请输入两个整数 :");  scanf("%d,%d",&a,&b);  p1=&a;  p2=&b;  change(p1,p2);  printf(" 输出互换后的两个整数 :");  printf("%d,%d\n",\*p1,\*p2);  }  void change(int \*pt1,int \*pt2)  {  int t; /\* 注意 :t 是整型变量 \*/  t=\*pt1;  \*pt1=\*pt2;  \*pt2=t;  }  程序运行结果：  请输入两个整数 :1,2 ↙  输出互换后的两个整数 :2,1  程序分析：  在本程序中，实际参数 p1 和 p2 与形式参数 pt1 和 pt2 均为指针变量。在函数调用时，实参 p1 和 p2 将它们的值（变量 a 和 b 的地址）分别传给形式参数 pt1 和 pt2，此时，pt1 和 pt2 也指向了变量 a 和 b。在 change 函数中，借助整型变量 t 交换 pt1 和 pt2 所指变量的值，实际上就是交换了变量 a 和 b 的值。  **【学生】**思考、讨论。 | **教师通过指针做函数参数展示，让学生了解指针做函数参数的基本理论知识。** |
| **课堂小结**  （3min） | 【**教师**】**回顾和总结本节课的知识点。**  **这节课我们一起学习了指针做函数参数，让学生具有无论做什么事情，都能努力追求效率与质量的品质。** | 通过对所学知识的回顾，培养学生的归纳总结能力 |
| **作业布置**（2min） | 【**教师**】**布置课后作业**  **分析下列程序的功能，写出运行结果。注意与 program9\_8.c 的区别，得出必要结论。** | 通过课后练习，使学生巩固所学新知识 |
| **知识讲解**（45min） | **【教师】**指向函数的指针变量  **任务描述**  利用指向函数的指针求解两个数的大者。  **预备知识**  **一、指向函数的指针变量的定义**  一个函数总是占用一段连续的内存区，而函数名就是该函数占用内存区的首地址。  我们可以把函数的这个首地址（或称入口地址）赋予一个指针变量，使该指针变量指向该函数。然后通过指针变量就可以找到并调用这个函数。这种指向函数的指针变量称为“函数的指针变量”。  一个指向函数的指针变量的定义形式如下：类型标识符 (\* 指针变量名 )( );其中，“类型标识符”表示被指函数的返回值的类型。“（\* 指针变量名）”表示“\*”后面的变量是指针变量。最后的空括号表示指针变量所指的是一个函数。  **二、指向函数的指针变量的引用**  在定义了指向函数的指针变量之后，可以将一个函数的入口地址赋给它，这样就实现了这个指针指向了一个特定的函数。例如：  void fun();  void (\*p)();  p=fun;  这样，p 就指向了函数 fun。  在指针指向函数后，就可以通过调用指针来调用函数。调用的一般形式为：  （\* 指针变量）（实参表列）  例如：（\*p）（a，b）就相当于 fun（a，b）。  **任务实施**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* 程序 : program9\_9.c \*/  /\* 功能 : 利用指向函数的指针 , 求解两个数的大者。 \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  int max(int x,int y)  {  if(x>y) return x;  else return y;  }  void main()  {  int (\*pmax)(); /\* pmax 为函数指针变量 \*/  int a,b,c;  pmax=max; /\* pmax 指向被调函数的入口地址 \*/  printf("please input two numbers:");  scanf("%d,%d",&a,&b );  c=(\*pmax)(a,b); /\* 用函数指针变量形式调用函数 \*/  printf("max=%d\n",c);  }  程序运行结果：  please input two numbers:3,7 ↙  max=7  **【学生】**思考、讨论。 | **教师通过指向函数的指针变量展示，让学生了解指向函数的指针变量的基本理论知识。** |
| **课堂小结**  （3min） | 【**教师**】**回顾和总结本节课的知识点。**  **这节课我们一起学习了指向函数的指针变量，让学生能够用指针优化程序，在编程领域精益求精、勇于创新。。** | 通过对所学知识的回顾，培养学生的归纳总结能力 |
| **作业布置**（2min） | 【**教师**】**布置课后作业**  **简述指向函数的指针变量的定义。** | 通过课后练习，使学生巩固所学新知识 |
| **知识讲解**（45min） | **【教师】**指针型函数  **任务描述**  通过指针函数，输入 1~7 之间的一个整数，输出对应的星期名。  **预备知识**  一个函数的返回值可以是整型、实型、字符型等，同时它也可以返回一个指针型的数据。若如果一个函数的返回值为指针类型，则称为这个函数为指针型函数。  定义指针型函数的一般形式为：  类型标识符 \* 函数名 ( 形参表 )  {  　　函数体  }  其中函数名之前加了“\*”号，表明这是一个指针型函数 ，即返回值是一个指针。  类型标识符表示了返回指针值所指向的数据类型。  例如：  int \*fun()  {  函数体  }  表示 fun 是一个指针型函数，它返回的指针指向一个整型变量。  【注意】指向函数的指针变量和指针型函数这两者在写法和意义上的区别：  （1）“int （\*p）( )；”是一个变量定义，说明 p 是一个指向函数入口的指针变量，该指针变量所指函数的返回值是整型量，（\*p）两边的括号不能少。  （2）“int \*p( )；”则不是变量定义而是函数定义，说明 p 是一个指针型函数，其返回值是一个指向整型变量的指针，\*p 两边没有括号。作为函数说明，在括号内最好写入形式参数，这样便于与变量说明区别。  **任务实施**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* 程序 : program9\_10.c \*/  /\* 功能 : 利用指针函数 , 输入 1~7 之间的一个整数 , 输出对应的星期名。 \*/  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include <stdio.h>  void main()  {  int i;  char \*day\_name(int n); /\*day\_name 为指针型函数 \*/  printf(" 请输入 1-7 之间的整数 :");  scanf("%d",&i );  printf("%d 对应的星期名为 :%s\n",i,day\_name(i));  }  char \*day\_name(int n)  {  static char \*name[]={"Illeagl day", "Monday", "Tuesday",  "Wednesday", "Thursday", "Friday", "Saturday", "Sunday" };  return((n<1||n>7)?name[0]: name[n]);  }  程序运行结果：  请输入 1-7 之间的整数 :7 ↙  7 对应的星期名为 :Sunday  【说明】  （1）day\_name 是一个指针型函数，它的返回值指向一个字符串。该函数中定义了一个静态指针数组 name，用各个星期名及出错提示给数组 name 初始化赋值。（2）在主函数中，把输入的整数 i 作实参，在 printf 语句中调用 day\_name 函数并把 i 值传送给形参 n。day\_name 函数中的 return 语句包含一个条件表达式，n 值若大于 7或小于 1，则把 name[0] 指针返回主函数，输出错误提示，否则返回主函数，输出对应的星期名。  **【学生】**思考、讨论。 | **教师通过指针型函数展示，让学生了解指针型函数的基本理论知识。** |
| **课堂小结**  （3min） | 【**教师**】**回顾和总结本节课的知识点。**  **这节课我们一起学习了指针型函数，让学生能够灵活使用指针编写出精炼且高效的程序，在编程领域贡献力量。** | 通过对所学知识的回顾，培养学生的归纳总结能力 |
| **作业布置**（2min） | 【**教师**】**布置课后作业**  **输入 10 个整数，将其中最小的数与第一个数对换，把最大的数与最后一个数对换。写 3 个函数：（1）输入 10 个数；（2）进行处理；（3）输出 10 个数。** | 通过课后练习，使学生巩固所学新知识 |
| **教学反思** | 对教材中的重点和难点内容，做到心中有数，并经常了解学生的态度，知识基础，学习方法，增强教学的针对性。 | |