|  |  |
| --- | --- |
| # | 题目 |
| **1.** | 写出下述程序的输出结果\_\_\_\_\_\_\_\_\_\_  #include <stdio.h> main() {          int a[] = {8,0,3,7,2,5}, i, j, t;          i = 1;          while ( i<6 )          {              t = a[i];              j = i-1;              while( j>=0 && t>a[j] )             {                   a[j+1] = a[j];                   j--;             }             a[j+1] = t;             i++;          }         i = 0;         while(i<5)              printf("%d,",a[i++]);         printf("%d",a[i]); } |
| **2.** | 下面的函数itoh(n,s)完成将无符号十进制整数转换成十六进制表示，并存入字符串数组s中。程序中用到的函数reverse(char s[])是一个将字符串置逆的函数(见题后)。 void itoh(unsigned n,char s[]) {  int h,i=0;  do{   h=n%16;   s[i++]=(h<=9)?h+’0’:h+’A’\_\_\_\_\_;  }while((n/=16)!=0); \_\_\_\_;  reverse(s); } void reverse(char s[]) {  int temp,i,j;  for(i=0,j=strlen(s)-1;i<j;i++,j--)  {   temp=s[i];   s[i]=s[j];   s[j]=temp;  } }  \_\_\_\_\_\_\_\_\_\_(所有答案填写在此空内，答案间以空格隔开) |
| **3.** | 以下程序的运行结果是 \_\_\_\_\_\_\_\_    #include<stdio.h> main() {  int a[10]={1,2,3,4,5,6,7,8,9,10};  int k,s,i;  float ave;  for (k=s=i=0;i<10;i++)  {   if (a[i]%2!=0)    continue;   s+=a[i];   k++;  }  if (k!=0)  {   ave=s/k;   printf("%d,%.2f\n",k,ave);  } } |
| **4.** | 下面程序段的运行结果是  \_\_\_\_\_          char ch[ ] = "600"; int a, s = 0; for (a = 0; ch[a] >= ’0’ && ch[a] <= ’9’; a++)     s = 10 \* s + ch[a] - ’0’; printf("%d", s); |
| **5.** | 下面的函数invert的功能是将一个字符串的内容颠倒过来。\_\_\_\_\_\_\_\_\_(所有答案填写在此空内，答案间以空格隔开) void invert (char str [] ) {  int i,j,填空1\_\_\_\_\_\_;  for(i=0,j=strlen(str)填空2\_\_\_\_\_\_\_;i<j;i++,j--)  {   k=str[i];   str[i]=str[j];   str[j]=k;  } } |
| **6.** | 函数squeez(char s[],char c)的功能是删除字符串s中所出现的与变量c相同的字符。 squeez(char s[],char c) {  int i,j;  for(i=j=0;填空1\_\_\_\_\_\_;i++)   if(s[i]!=c)    填空2\_\_\_\_\_\_;  s[j]='\0'; }  \_\_\_\_\_\_\_\_(所有答案填写在此空内，答案间以空格隔开) |
| **7.** | 函数index(char s[],char t[])检查字符串s中是否包含字符串t，若包含，则返回t在s中的开始位置(下标值)，否则返回-1。 index(char s[],char t[]) {  int i,j,k;  for(i=0;s[i]!='\0';i++)  {   for(j=i,k=0;填空1\_\_\_\_\_\_&&s[j]==t[k];j++,k++)    ;   if(填空2\_\_\_\_\_\_\_)    return (i);  }  return (-1); }  \_\_\_\_\_\_\_\_\_\_\_(所有答案填写在此空内，答案间以空格隔开) |

|  |  |
| --- | --- |
|  |  |
| **1.** | 以下关于数组的描述正确的是 (A)数组的大小是固定的,但可以有不同类型的数组元素 (B)数组的大小是可变的,但所有数组元素的类型必须相同 (C)数组的大小是固定的,所有数组元素的类型必须相同 (D)数组的大小是可变的,可以有不同类型的数组元素 |
| **2.** | 对以下说明语句的正确理解是\_\_\_\_\_\_\_            int a[10] = {6, 7, 8, 9, 10}; (A)  将5个初值依次赋给a[1]至a[5] (B)  将5个初值依次赋给a[0]至a[4] (C)  将5个初值依次赋给a[6]至a[10] (D)  因为数组长度与初值的个数不相同,所以此语句不正确 |
| **3.** | 在定义 int a[10]; 之后,对a的引用正确的是\_\_\_\_\_\_\_ (A)a[10] (B)a[6.3] (C)a(6) (D)a[10-10] |
| **4.** | 若有以下程序段： int a[]={4,0,2,3,1},i,j,t; for(i=1;i<5;i++) {  t=a[i];j=i-1;  while(j>=0&&t>a[j])  { a[j+1]=a[j];j--;}  a[j+1]=t; } 则该程序段的功能是\_\_\_\_\_\_\_。 (A)对数组a进行插入排序(升序) (B)对数组a进行插入排序(降序) (C)对数组a进行选择排序(升序) (D)对数组a进行选择排序(降序) |
| **5.** | 下列关于文件的结论中正确的是\_\_\_\_\_\_\_  A.对文件的操作必须先关闭文件 B.对文件的操作必须先打开文件 C.对文件的操作顺序没有统一规定 D.以上三个答案全部错误 |
| **6.** | 若用数组名作为函数调用的实参,传递给形参的是\_\_\_\_\_\_  (A)  数组的首地址 (B)  数组第一个元素的值 (C)  数组中全部元素的值 (D)  数组元素的个数 |
| **7.** | 有下面的程序段： char a[3], b[] = "China"; a = b; printf("%s", a); 则   \_\_\_\_\_\_\_\_  (A)  运行后将输出China (B)  运行后将输出Ch (C)  运行后将输出Chi (D)  编译出错 |
| **8.** | C语言中，对文件操作的一般步骤是\_\_\_\_\_\_\_         A. 打开文件－>操作文件－>关闭文件  B. 操作文件－>修改文件－>关闭文件 C. 读写文件－>打开文件－>关闭文件  D. 读文件－>写文件－>关闭文件 |
| **9.** | 下列语句中，将c定义为文件型指针的是 \_\_\_\_\_        A.  FILE c;   B.  FILE \*c;    C.  file c;   D.  file \*c; |
| **10.** | 下面程序的功能是将字符串s中所有的字符’c’删除。请选择填空。 #include<stdio.h> main() {  char s[80];  int i,j;  gets(s);  for(i=j=0;s[i]!=’\0’;i++)   if(s[i]!=’c’)\_\_\_\_ ;  s[j]=’\0’;  puts(s); }  (A)s[j++]=s[i]; (B)s[++j]=s[i]; (C)s[j]=s[i];j++ (D)s[j]=s[i] |
| **11.** | 下述对C语言字符数组的描述中错误的是           (A)  字符数组可以存放字符串 (B)  字符数组的字符串可以整体输入,输出 (C)  可以在赋值语句中通过赋值运算符"="对字符数组整体赋值 (D)  不可以用关系运算符对字符数组中的字符串进行比较 |
| **12.** | 下面程序段的运行结果是 \_\_\_\_\_\_。(其中#表示空格) char a[7]="abcdef",b[4]="ABC"; strcpy(a,b); printf("%c",a[5]); (A)# (B)\0 (C)e (D)f |
| **13.** | 要打开一个已存在的非空文件“file”用于修改，正确的语句是: A.fp=fopen("file","r"); B.fp=fopen("file","a+"); C.fp=fopen("file","w"); D.fp=fopen("file","r+"); |