**第1题（15分）**

**上机要求：**按程序功能要求编写程序，并将程序代码**保存到 E:\answer\3-A-1.c**中。

**程序功能：**实验室技能比赛打分（分数为1~10分），评委分为专家评委（用1表示）和大众评委（用2表示），评委总数不超过10位。打分规则如下：专家评委和大众评委的分数先分别取一个平均分（double型），然后，选手得分 = 专家评委平均分 \* 0.6 +大众评委平均分 \* 0.4；如果没有大众评委，则选手得分 =专家评委平均分。选手得分为double型，专家评委不可缺少。

（1）输入评委数n（int型变量）；

（2）输入n个评委类别（int型一维数组）及相应的打分（int型一维数组）；

（3）求解并输出选手得分（double型变量，保留两位小数）。

**样例输入1：**

**5**

**1 7**

**2 8**

**1 8**

**2 9**

**1 6**

**样例输出1：**

**7.60**

**样例输入2：**

**5**

**1 8**

**1 9**

**1 8**

**1 7**

**1 10**

**样例输出2：**

**8.40**

**参考答案及评分说明，共15分**

|  |  |
| --- | --- |
| 参考程序 | 得分说明 |
| #include <stdio.h>  #define N 10  int main()  {  int i,n,judge\_type[N],score[N],expert=0,normal=0,expertnum=0;  double avg;  scanf("%d",&n);  for(i=0;i<n;i++)  {  scanf("%d%d",&judge\_type[i],&score[i]);  if(judge\_type[i] ==1)  {  expert += score[i];  expertnum++;  }  else  {  normal += score[i];  }  }  if(expertnum == n)  avg = expert\*1.0/n;  else  avg = expert\*1.0/expertnum\*0.6 + normal\*1.0/(n-expertnum)\*0.4;  printf("%.2f\n",avg);  return 0;  } | 程序框架：**2**分  输入评委数：**2**分  输入评委类别：**2**分  输入评委打分：**2**分  求解并输出选手得分：**7**分 |

**第2题（16分）**

**上机要求：**按程序功能要求编写程序，并将程序代码**保存到 E:\answer\3-A-2.c** 中。

**程序功能：**在情报传递过程中，为了防止情报被截获，往往需要对情报用一定的方式加密，简单的加密算法虽然不足以完全避免情报被破译，但仍然能防止情报被轻易的识别。我们给出一种加密方法，对给定的一个明文字符串（括号中是一个“明文 -> 密文”的例子）:

（1）明文中所有的字母都在字母表中被循环右移了三个位置（XYza ->ABcd），其他非字母字符不变；

（2）大小写反转（ABcd ->abCD）。

编写程序，输入明文字符串（含空格），输出加密后的密文字符串。

**输入：**

输入一行，包含一个字符串，长度小于80个字符。

**输出：**

输出加密字符串。

**样例输入:**

**Hello! XyLan 1001**

**样例输出:**

**kHOOR! aBoDQ 1001**

**参考答案及评分说明，共16分**

|  |  |
| --- | --- |
| 参考程序 | 得分说明 |
| #include <stdio.h>  #include <string.h>  #define N 81  void shift3\_right(char s[])//循环右移3个字母  {  int i;  for(i = 0; s[i]; i++) {  if(s[i]>='a'&&s[i]<='z')  s[i]=(s[i]+3-'a')%26+'a';  else if(s[i]>='A'&&s[i]<='Z')  s[i]=(s[i]+3-'A')%26+'A';  }  }  void dxzh(char s[])//大小写转换  {  int i;  for(i=0;s[i];i++)  if(s[i]>='a'&&s[i]<='z')  s[i]-=32;  else if(s[i]>='A'&&s[i]<='Z')  s[i]+=32;  }  int main()  {  static char str[N];  int i;  gets(str);  shift3\_right(str);  dxzh(str);  puts(str);  return 0;  } | 程序框架：**2**分  字母循环右移：**5**分  字母大小写转换并输出：**5**分  字符数组定义：**2**分  字符串输入：**2**分 |

**第3题（18分）**

**上机要求：**按程序功能要求编写程序，并将程序代码**保存到 E:\answer\3-A-3.c** 中。

**功能说明：**从键盘输入5行6列整型二维数组的各元素，然后将数组各行元素分别按降序排序，再按行输出数组的各元素。

（1）输入int型二维数组；

（2）将数组各行元素分别按降序排序；

（3）输出排序后的数组（分5行6列，使用“%4d”格式输出）。

**样例输入：**

**56 78 36 4 50 80**

**19 44 95 72 -8 60**

**85 67 -3 32 12 35**

**29 21 47 88 28 -9**

**7 66 53 40 20 15**

**样例输出：**

**80 78 56 50 36 4**

**95 72 60 44 19 -8**

**85 67 35 32 12 -3**

**88 47 29 28 21 -9**

**66 53 40 20 15 7**

**参考答案及评分说明，共18分**

|  |  |
| --- | --- |
| 参考程序 | 得分说明 |
| #include <stdio.h>  #define N 5  #define M 6  int main()  {  int a[N][M],i,j,k,t;  for(i=0; i<N; i++)  for(j=0 ;j<M; j++)  scanf("%d",&a[i][j]);  for(j=0; j<N; j++)  for(i=0; i<M-1; i++)  for(k=i+1; k<M; k++)  if(a[j][i]<a[j][k])  { t=a[j][i]; a[j][i]=a[j][k]; a[j][k]=t; }  for(i=0;i<N;i++)  { for(j=0;j<M;j++)  printf("%4d",a[i][j]);  printf("\n");  }  return 0;  } | 程序框架：**2**分  二维数组定义：**2**分  二维数组输入：**3**分  数组各行降序排序：**8**分  数组输出：**3**分 |