C语言程序设计大作业

**二十四点游戏**

单 位： 信息工程学院

班 级：计算机科学与技术1班

学 号： 2015551122

姓 名： 李清振

任课教师： 谢勇

湘 潭 大 学

2016年2月

1. **题目概述**
2. **概述**

该题目主要考察用一些基础的C知识来完成现实生活中的简单运算。我们知道，计算机本身是不可能懂得四则运算的，想要计算机高明，你必须比它更高明；就是由你去告诉它该怎样做。简单来说这个题目更像是编一个计算器。

1. **所要完成的任务说明**

|  |
| --- |
| 任务 |
| 编写24点游戏程序 |
| 游戏给出4个1-13的整数 |
| 用户输入一个含这四个整数的运算表达式（可以有括号）游戏判断用户的表达式是否得到24。 |
| 游戏给出的数必须保证有解 |

1. **完成情况**

|  |  |
| --- | --- |
| 任务 | 完成情况说明 |
| 编写24点游戏程序 | 除掉包含中括号运算的表达式，该程序均能准确计算出相应结果，可以完成对式子结果判断的任务。 |
| 游戏给出4个1-13的整数 | 程序利用C中的stdlib.h、time.h文件下rand()函数、time()函数得到范围为1-13的四个随机数。 |
| 游戏给出的数必须保证有解 | 程序算法已经想出，但未真正写入程序。 |

1. **开发环境说明**

编程工具：code blocks软件

使用语言：C语言

1. **程序源代码及设计说明书**

|  |
| --- |
| 设计说明书：  第一部分 游戏规则说明，用printf()函数打印出游戏规则，创建游戏界面；  第二部分 以time(0)为种子利用rand()函数获得四个随机的范围为1-13的双精度浮点数；  第三部分 获取计算表达式放置在字符型数组中，并将数字与四则运算符号分开放置到double型数组和char型数组中等待处理；调用自定义函数sort()给double型数组排序，同时给得到的四个随机数排序，再用for循环判断相等，是则往下进行，否则打印“重新输入”，使用goto跳转到获取表达式那一步；  第四部分 使用if语句判断字符型数组中有无括号，若无则调用自定义函数forluma()计算算式的结果并赋值给double型变量result；  第五部分 若有括号则先去掉括号，具体做法是使用for循环找出括号中的运算符和数字（顺序不能乱），再调用自定义函数forluma()计算出结果，该结果再与剩余元素组成数组，使用过的运算符去掉；再次使用if语句判断括号的有无，无则再次调用自定义函数forluma()计算结果并赋值给双精度变量result；  第六部分 若有括号，再去括号，使用自定义函数forluma()计算出结果组成新数组，最后再调用自定义函数forluma()得结果并赋值双精度变量result；  第七部分 以上程序就将不带中括号的表达式结果准确计算出来了，使用if语句判断双精度变量result的值是否为24，若是，printf()打印出“恭喜你，答案正确！”，否则打印“很遗憾，答案错误……”  第八部分 设计程序输入数字“1”继续游戏，“0”结束游戏；若为继续游戏，则使用time(0)为种子，rand()函数重新获得四个范围为1-13的随机数（双精度浮点数），goto语句跳转到获取表达式那一步；输入“0”直接结束程序。 |
| #include <stdio.h>  #include <math.h>  #include <string.h>  #include <time.h>  #include <stdlib.h>  double sort(double a[],double n);  double forluma(double figure[],char sign[],int q);  int main(void)  {  printf("-----------------二十四点游戏-----------------\n");  printf("游戏说明：\n游戏会给你四个数字（1--13）\n你需要利用（不可重复使用）这四个数字\n");  printf("运用四则运算（允许加括号）得到24的结果\n");  printf("若输入的数字不符合题目要求，游戏会提醒“输入答案”\n"  "输入的式子以最简形式为标准\n");  printf("----------------------------------------------\n");  //游戏介绍  int i,j,x=3,y=2,y1,cout,k,t1,t2=0,t3,p,countin;  double data[4],figure[4],copy[4],figurebox[4],result,m[4];  char start[3]={"YES"},start1[3],sign[7],signbox[4],answer[20],n[7];  printf("现在，请输入“YES”进入游戏\n");  gets(start1);  if(start1[0]=='Y'&&start1[1]=='E'&&start1[2]=='S'){//得到随机数，并放进名为data的数组内  srand(time(0));  printf("游戏开始\n");  printf("下面是你将用到的四个数字\n");  for(i=0;i<4;i++){  data[i]=1+rand()%13;  printf("%3g",data[i]);}  printf("\n请输入您的答案\n"); //得到随机数，并放进名为data的数组内  }  TOP2:gets(answer);  for(i=0;i<strlen(answer);i++){  if(answer[i]=='('||answer[i]==')')y++;  }y1=y;  for(i=strlen(answer)-1;i>=0;i--){  if(answer[i]>47&&answer[i]<58){  if((answer[i-1]>47&&answer[i-1]<58)&&i>0){  figure[x]=answer[i]-38;i--;  }  else  figure[x]=answer[i]-48;x--;}  else {  sign[y]=answer[i];y--;}  } //式子的拆分  for(i=0;i<4;i++)copy[i]=figure[i];  sort(data,4);sort(copy,4);  for(i=0,j=0;i<4;i++){  if(data[i]==copy[i])j++;  }  if(j==4){  for(i=0,j=0;i<=y1;i++){  if(sign[i]=='(')j=1;  }if(j!=1){  result=forluma(figure,sign,4);printf("=%g\n",result);//判断输入的数字是否符合要求  if(result==24)printf("恭喜你，答案正确！");  else printf("很遗憾，答案错误……");  }  else{  for(i=0,j=0,k=0;i<strlen(sign);i++){  if(sign[i]=='('){  t1=i;t3=i;  while(sign[t1]!=')'){  figurebox[j]=figure[t1];  signbox[k]=sign[t1+1];  t1++;j++;k++;}  if(sign[t1]==')')break;  }  }  for(i=k-1;i<strlen(signbox);i++)signbox[i]='\0';  figure[t3]=forluma(figurebox,signbox,j);  for(i=t3+1;i<t1;i++)figure[i]=1e100;  for(i=t3;i<=t1;i++)sign[i]='#';  for(i=0,j=0;i<4;i++){  if(figure[i]!=1e100){m[j]=figure[i];j++;}  }for(i=j;i<4;i++)m[i]=1;  for(i=0,k=0;i<=y1;i++){  if(sign[i]!='#'){n[k]=sign[i];k++;}  }for(i=0,t2=k;i<4-j;i++,t2++)n[t2]='+';n[t2-1]='\0';//去括号  for(i=0,p=0;i<k;i++){if(n[i]=='(')p=1;}  if(p==1){  for(i=0;i<t2;i++)sign[i]=n[i];sign[t2-1]='\0';  for(i=0;i<4;i++)figure[i]=m[i];  for(i=0,j=0,k=0;i<strlen(sign);i++){  if(sign[i]=='('){  t1=i;t3=i;  while(sign[t1]!=')'){  figurebox[j]=figure[t1];  signbox[k]=sign[t1+1];  t1++;j++;k++;}  if(sign[t1]==')')break;  }  }  for(i=k-1;i<strlen(signbox);i++)signbox[i]='\0';  figure[t3]=forluma(figurebox,signbox,j);  for(i=t3+1;i<t1;i++)figure[i]=1e100;  for(i=t3;i<=t1;i++)sign[i]='#';  for(i=0,j=0;i<4;i++){  if(figure[i]!=1e100){m[j]=figure[i];j++;}  }for(i=j;i<4;i++)m[i]=1;  for(i=0,k=0;i<=y1;i++){  if(sign[i]!='#'){n[k]=sign[i];k++;}  }for(i=0,t2=k;i<4-j;i++,t2++)n[t2]='+';n[t2-1]='\0';  result=forluma(m,n,2);  }  else {result=forluma(m,n,j);}  printf("=%g\n",result);  if(result==24)printf("恭喜你，答案正确！");  else printf("很遗憾，答案错误……");}//得到式子的结果，并输出  printf("\n继续游戏请输入“1”，结束游戏请输入“0”\n");  scanf("%d",&countin);  if(countin==1){  srand(time(0));  for(i=0;i<4;i++){  data[i]=1+rand()%13;  printf("%3g",data[i]);  }x=3;y=2;t2=0;  goto TOP2;}  }else {printf("\n请输入答案\n");goto TOP2;}  return 0;  }  double sort(double a[],double n)//数字排序函数  {  double canshu;  int sorti,sortj,sortm;  for(sorti=0,sortm=n-1;sorti<n-1;sorti++,sortm--){  for(sortj=0;sortj<sortm;sortj++){  if(a[sortj]>=a[sortj+1]){  canshu=a[sortj];  a[sortj]=a[sortj+1];  a[sortj+1]=canshu;  }  }  }  return 0;  }  double forluma(double figure[],char sign[],int q)//不带括号的式子的计算  {  double answer;  int i,j,cout=0;  for(i=0;i<strlen(sign);i++){  if(sign[i]=='/'){  sign[i]='\*';figure[i+1]=1/figure[i+1];//除号变乘号  }  }  TOP1:for(i=0;i<strlen(sign);i++){  if(sign[i]=='\*'){  answer=figure[i]\*figure[i+1];  figure[i]=answer;  figure[i+1]=1e100;  sign[i]='#';cout++;  break;  }  }  for(i=0,j=0;i<q;i++){  if(figure[i]!=1e100){figure[j]=figure[i];j++;}  }  for(i=0,j=0;i<strlen(sign);i++){  if(sign[i]!='#'){sign[j]=sign[i];j++;}  }  for(i=0;i<strlen(sign)-cout;i++){  if(sign[i]=='\*')goto TOP1;//计算乘法  }  for(i=0;i<strlen(sign)-cout;i++){  if(sign[i]=='-'){  sign[i]='+';figure[i+1]=0-figure[i+1];//减号变加号  }  }  for(i=0,answer=0;i<q-cout;i++)answer+=figure[i];//计算加法  return answer;  } |

1. **程序分析与设计**
2. **系统分析**

以下任意要求在文档中缺少，不得分

1. 创建的函数

|  |  |
| --- | --- |
| **//数字排序函数**  double sort(double a[],double n){  double canshu;  int sorti,sortj,sortm;  for(sorti=0,sortm=n-1;sorti<n-1;sorti++,sortm--){  for(sortj=0;sortj<sortm;sortj++){  if(a[sortj]>=a[sortj+1]){  canshu=a[sortj];  a[sortj]=a[sortj+1];  a[sortj+1]=canshu;  }  }  }  return 0; | 1. **类型：**double； 2. **处理对象：**数组； 3. **函数功能：**   给一个double类型数组中的元素进行由小到大排序（冒泡排序）；   1. **参数意义：**   ①a[] double类型数组，也是该函数的主要处理对象  ②n 数组a中元素个数  5、**返回值类型：**0 |
| **//不带括号的简单四则运算**  double forluma(double figure[],char sign[],int q)//不带括号的式子的计算  {  double answer;  int i,j,cout=0;  for(i=0;i<strlen(sign);i++){  if(sign[i]=='/'){  sign[i]='\*';figure[i+1]=1/figure[i+1];//除号变乘号  }  }  TOP1:for(i=0;i<strlen(sign);i++){  if(sign[i]=='\*'){  answer=figure[i]\*figure[i+1];  figure[i]=answer;  figure[i+1]=1e100;  sign[i]='#';cout++;  break;  }  }  for(i=0,j=0;i<q;i++){  if(figure[i]!=1e100){figure[j]=figure[i];j++;}  }  for(i=0,j=0;i<strlen(sign);i++){  if(sign[i]!='#'){sign[j]=sign[i];j++;}  }  for(i=0;i<strlen(sign)-cout;i++){  if(sign[i]=='\*')goto TOP1;//计算乘法  }  for(i=0;i<strlen(sign)-cout;i++){  if(sign[i]=='-'){  sign[i]='+';figure[i+1]=0-figure[i+1];//减号变加号  }  }  for(i=0,answer=0;i<q-cout;i++)answer+=figure[i];//计算加法  return answer;  } | 1. **函数功能：**   支持多位的不带括号的简单的四则运算，精度小数点后六位；   1. **处理对象：**   double型和char型数组   1. **参数意义：**   ①figure[] double型数组，包含需处理的数字，该函数主要处理对象之一  ②sign[] 字符型数组包含四则运算符号，该函数主要处理对象之一  ③q 整数型，表示数组figure中元素的个数  4、**返回值类型：**double |
|  |  |

函数之间的调用关系图

sort()→main()←forluma()

1. **函数设计**

要求：

在每个3级标题下，对创建的每个函数分别画出流程图，贴出对应代码和注释，无流程图、无代码注释、注释与流程图对应不上不得分。

|  |  |
| --- | --- |
| 代码及注释 | 流程图 |
| double sort(double a[],double n)//数字排序函数  {  double canshu;  int sorti,sortj,sortm;  for(sorti=0,sortm=n-1;sorti<n-1;sorti++,sortm--){  for(sortj=0;sortj<sortm;sortj++){  if(a[sortj]>=a[sortj+1]){  canshu=a[sortj];  a[sortj]=a[sortj+1];  a[sortj+1]=canshu;  }  }  }  return 0;  } | sort()函数流程图 |
| double forluma(double figure[],char sign[],int q)//不带括号的式子的计算  {  double answer;  int i,j,cout=0;  for(i=0;i<strlen(sign);i++){  if(sign[i]=='/'){  sign[i]='\*';figure[i+1]=1/figure[i+1];//除号变乘号  }  }  TOP1:for(i=0;i<strlen(sign);i++){  if(sign[i]=='\*'){  answer=figure[i]\*figure[i+1];  figure[i]=answer;  figure[i+1]=1e100;  sign[i]='#';cout++;  break;  }  }  for(i=0,j=0;i<q;i++){  if(figure[i]!=1e100){figure[j]=figure[i];j++;}  }  for(i=0,j=0;i<strlen(sign);i++){  if(sign[i]!='#'){sign[j]=sign[i];j++;}  }  for(i=0;i<strlen(sign)-cout;i++){  if(sign[i]=='\*')goto TOP1;//计算乘法  }  for(i=0;i<strlen(sign)-cout;i++){  if(sign[i]=='-'){  sign[i]='+';figure[i+1]=0-figure[i+1];//减号变加号  }  }  for(i=0,answer=0;i<q-cout;i++)answer+=figure[i];//计算加法  return answer;  } | forluma()函数流程图 |

1. **测试分析**
2. **测试与调试**

调试时发现很多错误，多是一些格式说明符与类型不符的问题，也有编辑的函数存在的漏洞，比如在计算加减法的函数中起先就没有注意到处理元素个数的问题，导致有些数据的处理出现错误，后来经过多次测试，漏洞得以修复。

1. 测试结果分析

对于执行单次的程序，程序可以胜任，不过在后期加入goto跳转语句时，出现问题（偶尔会出现计算错误）,还有循环时不能处理输入错误时的情况（一旦在循环中出现输入错误的情况，程序进入死循环，无法跳出不得不强行停止）。

1. 结论

编辑的程序，结果差强人意，有些要求没有做到，发现想和做根本是两码事，不过在做之前非常有必要理清思绪有一个清晰的算法，应当多去思考更简便的算法，但也不能一味地追求简便，毕竟也要学会处理困难复杂的情况，这在以后的发展中会是时常遇到的，应去努力适应。

自我总结自己平时对编程的付出很少，尤其是第一学期后半段，编程题目基本没有动过，落下很多知识。这次的大作业，还是收获很多的，以前理解不到位的问题摸清了思路，曾经未触及的知识也多少有了认识，了解到几种新的函数，学到了一些处理问题的技巧。但是这并不能说是我最大的收获，这次大作业练习让我真正意识到自负是多麽的可悲，要走的路还很长要做的学问还很多，编程不避免不了复杂繁琐，我需要做的只是迎难而上,克服懒惰，平常心对待。

1. **设计体会**
2. **设计过程中的问题及其解决方案**

一个比较大的漏洞出现在自定义的四则运算函数上，起先设计是硬性规定处理数小于等于四，编辑完成后调用至主函数发现没问题就继续进行下面步骤的编辑了，然而在后面的运行中该漏洞的严重性越来越明显，所以必须从新写函数。我在函数中又加入一个整形变量，来控制处理元素的数量，然后调入主函数，程序得以正常运行。

问题：函数处理元素个数不确定；

解决方案：函数中加入一个整形变量，控制循环以达到对处理元素个数的限制。

1. **设计过程中的错误及其原因分析**

错误：带中括号运算的算式不能成功计算

原因分析：程序中只要发现前括号就把后面的符号放入数组，直到发现第一个后括号，所以处理带中括号的式子时会丢掉一些数据，不能准确完成计算。

1. **体会和收获**

编程不是一件简单的事，看似简单的问题要编程解决的话可能会很复杂。计算机与人的思维几乎完全不同，所以学会编程必须要了解计算机思维，掌握些常用的编程技巧活学活用、举一反三。还有重要的一点，C语言函数库中有些函数的功能可以简化程序，所以了解C库函数也是很有必要的。

若要说收获，知识技能上一定是多少了解到点不过这并不能说是重点，收获必须得是能让自己改变点什么，而且是往好了变。

编完这个上百行的程序，当中遇到了很多困难，当遇见那种很复杂的问题，有思路但很复杂不愿去编，总想着会不会有简单的办法去解决。当然可能会有，想过了锻炼了思维，不过无论如何总会遇到那种不得不用很麻烦的办法来解决问题的情况，所以遇事不要怕，做事不要懒，多锻炼这种事情对以后总会有帮助的。很多事情不是难而是自己缺少一个正确看待问题的态度。

端正态度，迎难而上，不要看现在有多苦，想想未来有多甜；努力不一定成功，但不努力一定不成功，对未来多一点希望，心态好了自信就来了，成功自然越来越近。

我要做的是：心无旁骛，做一科专一科。

1. **自我评价**

及格

1. **学术诚信承诺书**

