PROBLEMS

# 1.

Write a program that will compare two dates (day, month, year) and will compute which is the bigger one.

# 2.

Write a program that will compare two dates (day, month, year) and will compute the difference in days between two. Define a struct date.

# 3.

Write a program that will compute the vector and scalar product of two vectors. Vectors are represented with coordinates in three-dimensional coordinate system. Define a struct vector.

# 4.

Write a struct for representing complex numbers. Then implement functions for addition, subtraction and multiplication of two complex numbers. Test the functions in a main program where you read two complex numbers from standard input.

# 5.

Read from standard input data for unknown number of students (not more then 100). Each row of the data is in following format:

• first name

• last name

• ID number (format xxyzzzz)

• four numbers (points for each problem) separated with tab space.

Write a program that will print list of students, where each row will have: last name, first name, number, and total points sorted by the number of points. BTW the names should be printed with first capital letter.

# 6.

Write a program that will read from standard input data for countries and will print on the standard output the name of the president of the country whose capital has largest population.

• Data for country: name, president, capital and population.

• Data for city: name and population.

• Data for president: name, political party

SOLUTIONS

# 1.

#include <stdio.h>  
  
*typedef struct* Date{  
 *int* day,month,year;  
}date;  
  
date readDate(){  
 date **date**;  
 scanf("%d.%d.%d",&**date**.day,&**date**.month,&**date**.year);  
 *return* **date**;  
}  
  
*void* printDate(date date){  
 printf("%.2d.%.2d.%.2d",date.day,date.month,date.year);  
}  
  
*int* compare(*int* int1, *int* int2){  
 *if*(int1>int2){  
 *return* 1;  
 } *else if*(int2>int1){  
 *return* -1;  
 } *else*{  
 *return* 0;  
 }  
}  
  
*int* datecmp(date date1, date date2){  
 *int* **cmpYear**= compare(date1.year,date2.year);  
 *if*(**cmpYear**!=0){  
 *return* **cmpYear**;  
 } *else*{  
 *int* **cmpMonth**= compare(date1.month,date2.month);  
 *if*(**cmpMonth**!=0){  
 *return* **cmpMonth**;  
 } *else*{  
 *return* compare(date1.day,date2.day);  
 }  
 }  
}  
  
*int* main(){  
 date **date1**=readDate();  
 date **date2**=readDate();  
 printDate(**date1**);  
 *if*(datecmp(**date1**,**date2**)==1){  
 printf(" > ");  
 } *else if*(datecmp(**date1**,**date2**)==-1){  
 printf(" < ");  
 } *else*{  
 printf(" == ");  
 }  
 printDate(**date2**);  
 *return* 0;  
}

# 2.

#include <stdio.h>  
  
*typedef struct* Date{  
 *int* day,month,year;  
}date;  
  
date readDate(){  
 date **date**;  
 scanf("%d.%d.%d",&**date**.day,&**date**.month,&**date**.year);  
 *return* **date**;  
}  
  
*void* printDate(date date){  
 printf("%.2d.%.2d.%.2d",date.day,date.month,date.year);  
}  
  
*int* compare(*int* int1, *int* int2){  
 *if*(int1>int2){  
 *return* 1;  
 } *else if*(int2>int1){  
 *return* -1;  
 } *else*{  
 *return* 0;  
 }  
}  
  
*int* datecmp(date date1, date date2){  
 *int* **cmpYear**= compare(date1.year,date2.year);  
 *if*(**cmpYear**!=0){  
 *return* **cmpYear**;  
 } *else*{  
 *int* **cmpMonth**= compare(date1.month,date2.month);  
 *if*(**cmpMonth**!=0){  
 *return* **cmpMonth**;  
 } *else*{  
 *return* compare(date1.day,date2.day);  
 }  
 }  
}  
  
*int* days(date date1, date date2){  
 *int* **days**=date1.day-date2.day;  
 **days**+=(date1.month-date2.month)\*30;  
 **days**+=(date1.year-date2.year)\*365;  
 *return* **days**;  
}  
  
*int* main(){  
 date **date1**=readDate();  
 date **date2**=readDate();  
 *if*(datecmp(**date1**,**date2**)==0){  
 printf("The dates are equal");  
 *return* 0;  
 }  
 *if*(datecmp(**date1**,**date2**)==1){  
 printDate(**date1**);  
 printf(" is greater than ");  
 printDate(**date2**);  
 printf(" by %d days.", days(**date1**,**date2**));  
 } *else*{  
 printDate(**date2**);  
 printf(" is greater than ");  
 printDate(**date1**);  
 printf(" by %d days.", days(**date2**,**date1**));  
 }  
 *return* 0;  
}

# 3.

#include <stdio.h>  
  
*typedef struct* Vector{  
 *float* x,y,z;  
} vector;  
  
vector readVector(){  
 vector **Vector**;  
 scanf("%f %f %f",&**Vector**.x,&**Vector**.y,&**Vector**.z);  
 *return* **Vector**;  
}  
  
*float* scalarProduct(vector vector1,vector vector2){  
 *return* vector1.x\*vector2.x+vector1.y\*vector2.y+vector1.z+vector2.z;  
}  
  
vector vectorProduct(vector v1, vector v2){  
 vector **v**;  
 **v**.x = v1.y \* v2.z - v1.z \* v2.y;  
 **v**.y = v1.z \* v2.x - v1.x \* v2.z;  
 **v**.z = v1.x \* v2.y - v1.y \* v2.x;  
 *return* **v**;  
}  
  
*int* main(){  
 vector **vector1**=readVector();  
 vector **vector2**=readVector();  
 vector **Vector**= vectorProduct(**vector1**,**vector2**);  
 printf("v1 \* v2 = %.2f\n", scalarProduct(**vector1**,**vector2**));  
 printf("v1 x v2 = [%.2f , %.2f , %.2f]",**Vector**.x,**Vector**.y,**Vector**.z);  
 *return* 0;  
}

# 4.

#include <stdio.h>  
  
*typedef struct* complexNumber{  
 *float* real,imaginary;  
} complex;  
  
complex readComplexNumber(){  
 complex **z**;  
 scanf("%f + %fi",&**z**.real,&**z**.imaginary);  
 *return* **z**;  
}  
  
*void* print(complex z){  
 printf("%.2f + %.2f\*i",z.real,z.imaginary);  
}  
  
*void* sum(complex z1, complex z2){  
 complex **c**;  
 **c**.real=z1.real+z2.real;  
 **c**.imaginary=z1.imaginary+z2.imaginary;  
 print(z1);  
 printf(" + ");  
 print(z2);  
 printf(" = ");  
 print(**c**);  
 printf("\n");  
}  
  
*void* subtraction(complex z1, complex z2){  
 complex **c**;  
 **c**.real=z1.real-z2.real;  
 **c**.imaginary=z1.imaginary-z2.imaginary;  
 print(z1);  
 printf(" - ");  
 print(z2);  
 printf(" = ");  
 print(**c**);  
 printf("\n");  
}  
  
*void* product(complex z1, complex z2){  
 complex **c**;  
 **c**.real=z1.real\*z2.real;  
 **c**.imaginary=z1.imaginary\*z2.imaginary;  
 print(z1);  
 printf(" \* ");  
 print(z2);  
 printf(" = ");  
 print(**c**);  
 printf("\n");  
}  
  
*int* main(){  
 complex **z1**=readComplexNumber();  
 complex **z2**=readComplexNumber();  
 sum(**z1**,**z2**);  
 subtraction(**z1**,**z2**);  
 product(**z1**,**z2**);  
 *return* 0;  
}

# 5.

#include <stdio.h>  
#include <ctype.h>  
#include <string.h>  
  
*typedef struct* student{  
 *char* name[20];  
 *char* surname[20];  
 *char* index[7];  
 *int* points[4];  
}student;  
  
student readStudent(){  
 student **s**;  
 scanf("%s %s %s",**s**.name,**s**.surname,**s**.index);  
 *for*(*int* **i**=0 ; **i**<4 ; **i**++){  
 scanf("%d",&**s**.points[**i**]);  
 }  
 *return* **s**;  
}  
  
*void* nameSurname(*char* \*name,*char* \*surname){  
 name[0]= toupper(name[0]);  
 *for*(*int* **i**=1; **i**< strlen(name) ; **i**++){  
 name[**i**]= tolower(name[**i**]);  
 }  
 surname[0]= toupper(surname[0]);  
 *for*(*int* **i**=1; **i**< strlen(surname) ; **i**++){  
 surname[**i**]= tolower(surname[**i**]);  
 }  
}  
  
*int* totalPoints(student s){  
 *int* **sum**=0;  
 *for*(*int* **i**=0 ; **i**<4 ; **i**++){  
 **sum**+=s.points[**i**];  
 }  
 *return* **sum**;  
}  
  
*int* grade(*int* points){  
 *int* **grade**=points/10+1;  
 *if* (**grade** > 10) {  
 **grade** = 10;  
 }  
 *if* (**grade** < 5) {  
 **grade** = 5;  
 }  
 *return* **grade**;  
}  
  
*int* studentcmp(student s1, student s2){  
 *if*(totalPoints(s1)> totalPoints(s2)){  
 *return* 1;  
 } *else if*(totalPoints(s1)< totalPoints(s2)){  
 *return* -1;  
 } *else*{  
 *return* 0;  
 }  
}  
  
*void* swap(student \*s1, student \*s2){  
 student **tmp**=\*s1;  
 \*s1=\*s2;  
 \*s2=**tmp**;  
}  
  
*void* bubbleSort(student s[], *int* n){  
 *for*(*int* **i**=0 ; **i**<n ; **i**++){  
 *for*(*int* **j**=0 ; **j**<n-1-**i** ; **j**++){  
 *if*(studentcmp(s[**j**],s[**j**+1])==-1){  
 swap(&s[**j**],&s[**j**+1]);  
 }  
 }  
 }  
}  
  
*void* printStudent(student s){  
 *int* **points**= totalPoints(s);  
 *int* **gr**= grade(**points**);  
 nameSurname(s.name,s.surname);  
 printf("Name: %s, Surname: %s, Index: %s, Points: %d, Grade: %d\n",s.name,s.surname,s.index,**points**,**gr**);  
}  
  
*int* main(){  
 *int* **n**;  
 student **STUDENTS**[100];  
 scanf("%d",&**n**);  
 *for*(*int* **i**=0 ; **i**<**n** ; **i**++){  
 **STUDENTS**[**i**]=readStudent();  
 }  
 bubbleSort(**STUDENTS**,**n**);  
 *for*(*int* **i**=0 ; **i**<**n** ; **i**++){  
 printStudent(**STUDENTS**[**i**]);  
 }  
 *return* 0;  
}

# 6.

#include <stdio.h>  
  
*typedef struct* City{  
 *char* name[20];  
 *int* population;  
}capital;  
  
*void* printCapital(capital c){  
 printf("%s %d ",c.name,c.population);  
}  
  
*typedef struct* President{  
 *char* name[20], party[10];  
}president;  
  
*void* printPresident(president p){  
 printf("%s %s ",p.name,p.party);  
}  
  
*typedef struct* Country{  
 *char* name[20];  
 president president;  
 capital capital;  
 *int* population;  
}country;  
  
*void* printCountry(country c){  
 printf("%s ",c.name);  
 printPresident(c.president);  
 printCapital(c.capital);  
 printf("%d\n",c.population);  
}  
  
capital readCapital(){  
 capital **c**;  
 scanf("%s %d",**c**.name,&**c**.population);  
 *return* **c**;  
}  
  
president readPresident(){  
 president **p**;  
 scanf("%s %s",**p**.name,**p**.party);  
 *return* **p**;  
}  
  
country readCountry(){  
 country **country**;  
 scanf("%s",&**country**.name);  
 **country**.president=readPresident();  
 **country**.capital=readCapital();  
 scanf("%d",&**country**.population);  
 *return* **country**;  
}  
  
*int* countryCmp(country c1, country c2){  
 *return* c1.capital.population>c2.capital.population;  
}  
  
*void* printMax(country max){  
 printf("%s",max.president.name);  
}  
  
*int* main(){  
 *int* **n**;  
 scanf("%d",&**n**);  
 *int* **flag**=1;  
 country **max**;  
 *for*(*int* **i**=0 ; **i**<**n** ; **i**++){  
 country **COUNTRY**=readCountry();  
 printCountry(**COUNTRY**);  
 *if*(**flag**){  
 **max**=**COUNTRY**;  
 **flag**=0;  
 } *else if*(countryCmp(**COUNTRY**,**max**)==1){  
 **max**=**COUNTRY**;  
 }  
 }  
 printf("The president of the country whose capital has largest population: ");  
 printMax(**max**);  
 //printf("%s",max.president.name);  
 *return* 0;  
}