/\***Q.1**

**Create a structure to store data of 100 students. The data to be stored is**

**name, contact\_number, roll no, marks, date of birth and date of admission**

**(where date of admission and date of birth have three members i.e. day, month and year).**

**Display details of students, having percentage greater than 80.**

\*/

#include<stdio.h>

struct school

{

char name[100];

char pno[10];

int rno;

int marks;

char dob[12];

char doa[12];

};

int main()

{

int n,i;

printf("Enter the number of students : ");

scanf("%d",&n);

struct school t[n];

char ch;

for(i=0;i<n;i++)

{

printf("Enter the name of student : ");

ch = getchar();

scanf("%[^\n]%\*c",t[i].name);

printf("Enter the Contact Number : ");

scanf("%[^\n]%\*c",&t[i].pno);

printf("Enter the Roll Number : ");

scanf("%d",&t[i].rno);

printf("Enter the Marks : ");

scanf("%d",&t[i].marks);

printf("Enter the Date of birth : ");

scanf("%s",t[i].dob);

printf("Enter the Date of admission : ");

scanf("%s",t[i].doa);

printf("\n");

}

for(i=0;i<n;i++)

{

if((t[i].marks)/5 >=80)

{

printf("\nName of student : %s",t[i].name);

printf("\nContact Number : %s",t[i].pno);

printf("\nRoll Number : %d",t[i].rno);

printf("\nMarks : %d",t[i].marks);

printf("\nPercentage : %d ",t[i].marks/5);

printf("\nDate of birth : %s",t[i].dob);

printf("\nDate of admission : %s",t[i].doa);

}

}

return 0;

}

/\***Q2.**

**Write a c program to compare two given dates using function named compare ().**

**And print following messages in the main function based on returned value by the function:**

**a) “dates are equal”, if returned value is 0.**

**b) “date1 is greater than date2”, if returned value is 1.**

**c) “date1 is smaller than date2”, if returned value is -1.**

\*/

#include<stdio.h>

struct Date

{

int date;

int month;

int year;

};

int compare()

{

struct Date d1,d2;

printf("Enter first date(dd mm yyyy):");

scanf("%d%d%d",&d1.date,&d1.month,&d1.year);

printf("\nEnter second date(dd mm yyyy):");

scanf("%d%d%d",&d2.date,&d2.month,&d2.year);

if((d1.date==d2.date)&&(d1.month==d2.month)&&(d1.year==d2.year))

return 0;

else if((d1.year>d2.year) || (d1.year==d2.year && d1.month > d2.month) || (d1.year == d2.year && d1.month == d2.month && d1.date > d2.date))

return 1;

else

return -1;

}

int main()

{

int rtrn=compare();

if(rtrn==0)

printf("\nDate 1 and Date 2 are equal");

else if (rtrn==1)

printf("\nDate 1 is greater than Date 2 ");

else

printf("\nDate 1 is smaller than Date 2");

return 0;

}

/\***Q3.**

**Write a c program to calculate difference between two time periods using structures.**

\*/

#include <stdio.h>

struct time

{

int s;

int m;

int h;

};

void timediff(struct time t1, struct time t2, struct time \*diff)

{

if(t2.s > t1.s)

{

--t1.m;

t1.s += 60;

}

diff->s = t1.s - t2.s;

if(t2.m > t1.m)

{

--t1.h;

t1.m += 60;

}

diff->m = t1.m - t2.m;

diff->h = t1.h - t2.h;

}

int main()

{

struct time t1, t2, diff;

printf("Enter time 1 (hours, minutes and seconds ): ");

scanf("%d%d%d", &t1.h, &t1.m, &t1.s);

printf("Enter time 2 (hours, minutes and seconds ): ");

scanf("%d %d %d", &t2.h, &t2.m, &t2.s);

timediff(t1, t2, &diff);

printf("\nTime Difference: %d:%d:%d - %d:%d:%d",t1.h,t1.m,t1.s,t2.h,t2.m,t2.s);

printf(" = %d:%d:%d\n", diff.h, diff.m, diff.s);

return 0;

}

/\***Q4.**

**Write a c program to add two distances (in inch-feet system) using structures.**

\*/

#include<stdio.h>

struct dist

{

int feet;

float inch;

} d1, d2, sum;

int main()

{

printf("Enter details for 1st distance\n");

printf("Enter feet: ");

scanf("%d", &d1.feet);

printf("Enter inch: ");

scanf("%f", &d1.inch);

printf("\nEnter details for 2nd distance\n");

printf("Enter feet: ");

scanf("%d", &d2.feet);

printf("Enter inch: ");

scanf("%f", &d2.inch);

sum.feet=d1.feet+d2.feet;

sum.inch=d1.inch+d2.inch;

if (sum.inch>12.0)

{

sum.inch-=12.0;

++sum.feet;

}

printf("\nSum of distances = %d' %.1f\"", sum.feet,sum.inch);

return 0;

}

/\***Q5.**

**Create a structure to store data of 100 customers in a bank. The data to be stored**

**is account number, Name, Balance in the account. Then**

**a. Write a function to print account number and name of each customer with a balance<1,00,000.**

**b. Write a function to perform withdrawal and deposit. If customer request for withdrawal**

**or deposit, it is given in the form: account number, amount, code**

**(1 for deposit and 0 for withdrawal)**

**Write a c program to print a message, “The balance is insufficient for the specified withdrawal”.**

\*/

#include <stdio.h>

struct bank

{

char name[100];

int accno;

int bal;

}t[100];

void minbal(int n)

{

for(int i=0;i<n;i++)

if(t[i].bal<100000)

{ printf("\nDetails of bank holders having balance less than 1,00,000");

printf("\nName of bank holder : %s \nAccount number : %d \n",t[i].name,t[i].accno);

}

}

void withdraw(int i,int amt)

{

if(t[i].bal<amt)

printf("\nInsufficient Balance");

else

t[i].bal-=amt;

printf("Updated Details :\nName of bank holder : %s \nAccount Balance : %d \n",t[i].name,t[i].bal);

}

void deposit(int i,int amt)

{

t[i].bal+=amt;

printf("Updated Details :\nName of bank holder : %s \nAccount Balance : %d \n",t[i].name,t[i].bal);

}

int main()

{

int n,i;

printf("Enter the number of customers : ");

scanf("%d",&n);

char ch;

for(i=0;i<n;i++)

{

printf("Enter the name of bank holder : ");

ch= getchar();

scanf("%[^\n]%\*c",t[i].name);

printf("Enter the account number : ");

scanf("%ld",&t[i].accno);

printf("Enter the balance amount : ");

scanf("%ld",&t[i].bal);

}

minbal(n);

int code,amt,acno;

printf("\nEnter Withdraw or Deposit Request Codes\n1 For Deposit.\n0 For Withdraw\nFormat:Account\_number Amount Code\n");

printf("Enter : ");

scanf("%d%d%d",&acno,&amt,&code);

for(i=0;i<n;i++)

{

if(t[i].accno==acno)

{

if(code==0)

withdraw(i,amt);

else if(code==1)

deposit(i,amt);

else

printf("\nInvalid code");

}

}

return 0;

}

/\***Q6.**

**Write a c program to add two complex numbers by passing structure to a function and result should be printed in main function.**

#include<stdio.h>

struct comp

{

float rl;

float img;

}c1,c2;

struct comp complexsum(struct comp \*a,struct comp \*b)

{

struct comp temp;

temp.rl = (\*a).rl + (\*b).rl;

temp.img = (\*(a)).img + (\*(b)).img;

return temp;

}

int main(){

struct comp result;

printf("Enter Complex Number 1: ");

scanf("%f%f",&c1.rl, &c1.img);

printf("Enter Complex Number 2: ");

scanf("%f%f",&c2.rl,&c2.img);

result = complexsum(&c1,&c2);

printf("The sum is %.2f + %.2fi\n", result.rl,result.img);

return 0;

}

/\***Q7.**

**Write a menu driven program that depicts the working of a library The menu options should be:**

**a. Call a function named addbookdetail() to add book details(title, author name, price, edition, quantity, accession number and flag indicating whether the book is issued or not). After addition of a book, number of books should be increased by 1.**

**b. Call a function named total\_cost() to display total cost of each different book.**

**c. Call a function named book\_sort() to display book details sorted title wise.**

**d. Call a function named book\_search() to list all books of a given author. If available, display all details. Otherwise display a message “Not Available”.**

**e. Call a function named book\_quantity() to display total number of books in library.**

**f. Call a function named issue\_book() to issue a book. If someone issue a book then number of books should be decreased by 1.**

**g. Call a function named dis\_book() to display the books in the order of accession number.**

**h. Call a function named list\_title() to display the title of a specified book.**

\*/

#include<stdio.h>

#include<string.h>

void add();

void tcost();

void sort();

void search();

void qty();

void issue();

void accorder();

void list();

struct lib

{

 int accno;

 char title[20];

 char author[20];

 int cost;

int edition;

}b[100];

int count;

int main()

{ int ch;

  printf("\n \t\tMENU\n");

printf("\n 1: To add book details\n");

  printf("\n 2: Total cost of each books\n");

  printf("\n 3: Book sort on the bases of title\n");

  printf("\n 4: Search book on the bases of Author\n");

  printf("\n 5: Total books quantity\n");

printf("\n 6: Issue a Book\n");

printf("\n 7: Sort on the bases of accesion no.\n");

printf("\n 8: Display book title\n");

printf("\n Enter the choice :");

  scanf("%d",&ch);

  switch(ch)

  {

   case 1:add();

    break;

   case 2:tcost();

    break;

   case 3:sort();

    break;

   case 4:search();

    break;

case 5:qty();

    break;

   case 6:issue();

    break;

   case 7:accorder();

    break;

   case 8:list();

    break;

  }

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void add()

{

 printf("\n enter the detail of book \n");

 printf("\n enter the book title=");

 gets(b[count].title);

 printf("\n enter the name of author=");

 gets(b[count].author);

 printf("\n enter the cost of book=");

 scanf("%d",&b[count].cost);

 printf("\n enter the edition of book=");

 scanf("%d",&b[count].edition);

 printf("\n enter accession number of book =");

 scanf("%d",&b[count].accno);

count++;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void tcost()

{

 int i,cnt=0;

 for(i=0;i<count;i++)

 {

   printf("\n%s:%d",b[i].title,b[i].cost);

cnt+=b[i].cost;

 }

 printf("\n Total cost of books : &d",cnt);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void sort()

{

int i,j,temp;

char name[30];

for(i=0;i<count;i++)

{

for(j=0;j<count-1;j++)

{

if(strcmp(b[j].title,b[j+1].title)>=0)

{

strcpy(name,b[j].title);

strcpy(b[j].title,b[j+1].title);

strcpy(b[j+1].title,name);

temp=b[j].accno;

b[j].accno=b[j+1].accno;

b[j+1].accno=temp;

}

}

}

for(i=0;i<count;i++)

printf("\n %s : %d",b[i].title,b[i].accno);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void search()

{

 int i,cnt=0;

 char name[20];

 printf("\n enter the name of author=");

 gets(name);

 for(i=0;i<count;i++)

 {

  if(strcmp(name,b[i].author)==0)

  {

   cnt++;

   printf("\n %d\n%s\n%s\n%d",b[i].accno,b[i].title,b[i].author,b[i].cost);

  }

 }

 if(cnt==0)

  printf("\n Not Available \n");

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void qty()

{

 printf("\n total no of book in library =%d",count);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void issue()

{

 int i,cnt=0;

 char name[20];

 printf("\n enter the book title to be issued=");

 gets(name);

 for(i=0;i<count;i++)

 {

  if(strcmp(name,b[i].title)==0)

  {

   cnt++;

   printf("\n yes this book is available and can be issued");

  }

 }

 if(cnt==0)

  printf("\n Not available issued to someone\n");

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void accorder()

{

int i,j,temp;

char name[30];

for(i=0;i<count;i++)

{

for(j=0;j<count-1;j++)

{

if(b[j].accno>b[j+1].accno)

{

temp=b[j].accno;

b[j].accno=b[j+1].accno;

b[j+1].accno=temp;

strcpy(name,b[j].accno);

strcpy(b[j].accno,b[j+1].accno);

strcpy(b[j+1].accno,name);

}

}

}

for(i=0;i<count;i++)

printf("\n %d : %s",b[i].accno,b[i].title);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void list()

{

 int i,cnt=0,no;

 printf("\n enter the accession number = ");

 scanf("%d",&no);

 for(i=0;i<count;i++)

 {

  if(no==b[i].accno)

  {

   cnt++;

   printf("\n %s\n%s\n%d",b[i].title,b[i].author,b[i].cost);

  }

 }

 if(cnt==0)

  printf("\n Not Available \n");

}

/\***Q8.**

**An automobile company has the serial number for engine parts starting from AA0 to FF9. Other characteristics of parts to be specified in a structure are: material, quantity and date of manufacture. Write a program to retrieve information on parts with serial numbers between BB1 and CC6.**

\*/

#include<stdio.h>

#include<string.h>

struct car

{

char serial[50],mat[50],dom[10];

int qty;

};

int main()

{

int n,i;

printf("Enter Number : ");

scanf("%d",&n);

struct car t[n];

for(i=0;i<n;i++)

{

printf("\nEnter Serial number : ");

scanf("%s",t[i].serial);

printf("\nEnter Material : ");

scanf("%s",t[i].mat);

printf("\nEnter Date of Manufacturing : ");

scanf("%s",t[i].dom);

printf("\nEnter Quantity : ");

scanf("%d",&t[i].qty);

}

for(i=0;i<n;i++)

{

if(strcmp(t[i].serial,"BB1")>=0 && strcmp(t[i].serial,"CC6")<=0)

{

printf("\nSerial number : %s",t[i].serial);

printf("\nMaterial : %s",t[i].mat);

printf("\nDate of Manufacturing : %s",t[i].dom);

printf("\nQuantity %d",t[i].qty);

}

}

return 0;

}

/\***Q9.**

**There is a structure called employee that holds information like employee code, name, date of joining. Write a c program to create an array of structure and enter some data into it. Then ask the user to enter the current date. Display the names of those employees whose tenure is 3 or more than 3 years according to the given current date.**

\*/

#include<stdio.h>

struct emp

{

char ecode[30],name[30];

int d,m,y;

};

int main()

{

int n;

printf("Enter number of Employees : ");

scanf("%d",&n);

struct emp t[n];

int i,x,y,z;

for(i=0;i<n;i++)

{

printf("\nEnter Employee Code : ");

scanf("%s",t[i].ecode);

printf("\nEnter Employee Name : ");

scanf("%s",t[i].name);

printf("\nEnter Date of joining (dd mm yyyy) : ");

scanf("%d%d%d",&t[i].d,&t[i].m,&t[i].y);

}

printf("\nEnter Current Date (dd mm yyyy) : ");

scanf("%d%d%d",&x,&y,&z);

for(i=0;i<n;i++)

if((z-t[i].y)>=3)

printf("\n%s",t[i].name);

return 0;

}

/\***Q10.**

**Write a c program to implement below scheme: There are five players from which the most valuable player (MVP) is to be chosen. Each player is to be judged by 3 judges, who would assigned a rank to each player. The player whose sum of rank is highest is chosen as MVP.**

\*/

#include<stdio.h>

struct point

{

int j1,j2,j3;

};

int main()

{

struct point t[5];

int i;

for(i=0;i<5;i++)

{

printf("\nEnter Points given to player %d by judge 1 : ",i+1);

scanf("%d",&t[i].j1);

printf("\nEnter Points given to player %d by judge 2 : ",i+1);

scanf("%d",&t[i].j2);

printf("\nEnter Points given to player %d by judge 3 : ",i+1);

scanf("%d",&t[i].j3);

}

int index,sum,mvp=t[0].j1+t[0].j2+t[0].j3;

for(i=0;i<5;i++)

{

sum = t[i].j1 + t[i].j2 + t[i].j3 ;

if(mvp<sum)

{

mvp=sum;

index = i;

}

}

printf("\nMVP is %d .",++index);

return 0;

}