#include<windows.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<math.h>

struct studcoll

{

char coll1[10];

char coll2[10];

char coll3[10];

char coll4[10];

};

struct pref

{

char cname[10];

char cbranch[10];

};

struct student

{

int roll;

int marks;

char names[100];

struct pref p[3];

struct pref q;

};

struct branch

{

char name[100];

int capacity;

};

struct college

{

char collname[100];

int rgno;

struct branch b[4];

};

void merge(struct student s[], int l, int m, int r)

{

int i, j, k;

int n1 = m - l + 1;

int n2 = r - m;

/\* create temp structures\*/

struct student L[n1], R[n2];

/\* Copy data to temp structures L[] and R[] \*/

for (i = 0; i < n1; i++)

L[i] = s[l + i];

for (j = 0; j < n2; j++)

R[j] = s[m + 1+ j];

i = 0;

j = 0;

k = l;

while (i < n1 && j < n2)

{

if (L[i].marks >= R[j].marks)

{

s[k] = L[i];

i++;

}

else

{

s[k] = R[j];

j++;

}

k++;

}

while (i < n1)

{

s[k] = L[i];

i++;

k++;

}

while (j < n2)

{

s[k] = R[j];

j++;

k++;

}

}

void Sort(struct student s[], int l, int r)

{

if (l < r)

{

int m = l+(r-l)/2;

Sort(s, l, m);

Sort(s, m+1, r);

merge(s, l, m, r);

}

}

void SetColor(int ForgC)

{

WORD wColor;

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_SCREEN\_BUFFER\_INFO csbi;

if(GetConsoleScreenBufferInfo(hStdOut, &csbi))

{

wColor = (csbi.wAttributes & 0xF0) + (ForgC & 0x0F);

SetConsoleTextAttribute(hStdOut, wColor);

}

return;

}

int binarysearch(const struct student s[], int l, int r, int x)

{

int i;

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

{

if(s[i].roll==x)

{

return i;

}

}

return -1;

}

void sort(struct student s[],int y,int x)

{

int i,j;

struct student temp;

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

{

for(j=0;j<x;j++)

{

if(s[j+1].marks>s[j].marks)

{

temp=s[j+1];

s[j+1]=s[j];

s[j]=temp;

}

}

}

}

int binarySearch(const struct student s[], int l, int r, int x)

{

if (r >= l)

{

int mid = l + (r - l) / 2;

if (s[mid].roll == x)

return mid;

if (s[mid].roll > x)

return binarySearch(s, l, mid - 1, x);

return binarySearch(s, mid + 1, r, x);

}

return -1;

}

int main()

{

int i,j,totcol,totstud;

int flag,x,y,k,ch,temp;

struct college c[20];

printf("\t\t\t\t\t\t\t\tWELCOME TO THE CET COUNCELLING PORTAL\n\n\n");

printf("College registration\n\n");

SetColor(10);

printf("Please follow the following instruction for registration\n\n");

SetColor(11);

printf("1.Enter CSE for Computer Science and Engg.\n2.Enter ECE for Eloctonics and communication Engg.\n");

SetColor(12);

printf("3.Enter EEE for Electrical and Eloctronics Engg.\n");

SetColor(13);

printf("4.Enter MEC for Mechinal Engg.\n5.Enter AAR for Autoation And Robotics\n");

SetColor(14);

printf("6.Enter CIV for Civil Engg.\n7.Enter CHE for Chemical Engg.\n");

SetColor(15);

printf("8.Enter AER for Aeronautical Engg.\n9.Enter AGR for Agriculture Engg.\n");

SetColor(18);

printf("10.Enter MIE for Minning Engg.\n");

SetColor(17);

printf("11.Enter BOC for Biochemical Engg.\n");

SetColor(18);

printf("12.Enter MEG for Metallurgical Engg.\n");

SetColor(19);

printf("13.Enter IIE for Industrial Engg.\n");

SetColor(20);

printf("13.Enter MAF for Manufacturing Engg.\n");

SetColor(21);

printf("14.Enter PET for Petroleun Engg.\n");

SetColor(22);

printf("15.Enter MAR for Marine Engg.\n");

SetColor(23);

printf("16.Enter PHE for Photonics Engg.\n");

SetColor(24);

printf("17.Enter NAT for Nanotechnology Engg.\n");

SetColor(25);

printf("18.Enter CER for Ceramics Engg.\n\n\n\n");

SetColor(15); //15 for white color. and 16 for black color

printf("\*\*Only 4 Engineering Branches out of 18 mentioned above are allowed.\n");

printf("\*\*First come first serve.. only 10 colleges are allowed to participate.\n");

printf("\*\*Giving invalid branch code will lead to rejection of that branch.\n\n");

FILE \*fp=fopen("college.txt","r");

i=0;

while(!feof(fp))

{

fscanf(fp,"%s %d %s %d %s %d %s %d %s %d\n",c[i].collname,&c[i].rgno,c[i].b[0].name,&c[i].b[0].capacity,c[i].b[1].name,&c[i].b[1].capacity,c[i].b[2].name,&c[i].b[2].capacity,c[i].b[3].name,&c[i].b[3].capacity);

i++;

}

fclose(fp);

totcol=i;

struct student s[1000];

fp==fopen("student.txt","r");

i=0;

while(!feof(fp))

{

fscanf(fp,"%d %d %s %s %s %s %s %s %s",&s[i].roll,&s[i].marks,s[i].names,s[i].p[0].cname,s[i].p[0].cbranch,s[i].p[1].cname,s[i].p[1].cbranch,s[i].p[2].cname,s[i].p[2].cbranch);

i++;

}

fclose(fp);

totstud=i;

x=0;

while(x!=1)

{

SetColor(34);

printf("1.Want to add a college details . PRESS 1\n");

SetColor(20);

printf("2.Want to add a student details . PRESS 2\n");

SetColor(10);

printf("3.To end student and college registration. PRESS 3\n");

scanf("%d",&ch);

switch(ch)

{

case 1:i=totcol;

SetColor(17);

printf("\n\ncollege name\n");

SetColor(15);

scanf("%s",c[i].collname);

SetColor(18);

printf("Enter 5 digit unique registration no.\n");

SetColor(15);

scanf("%d",&c[i].rgno);

SetColor(19);

printf("Enter branch 1 name\n");

SetColor(15);

scanf("%s",c[i].b[0].name);

SetColor(19);

printf("Enter its capacity\n");

SetColor(15);

scanf("%d",&c[i].b[0].capacity);

SetColor(20);

printf("Enter branch 2 name\n");

SetColor(15);

scanf("%s",c[i].b[1].name);

SetColor(20);

printf("Enter its capacity\n");

SetColor(15);

scanf("%d",&c[i].b[1].capacity);

SetColor(21);

printf("Enter branch 3 name\n");

SetColor(15);

scanf("%s",c[i].b[2].name);

SetColor(21);

printf("Enter its capacity\n");

SetColor(15);

scanf("%d",&c[i].b[2].capacity);

SetColor(22);

printf("Enter branch 4 name\n");

SetColor(15);

scanf("%s",c[i].b[3].name);

SetColor(22);

printf("Enter its capacity\n");

SetColor(15);

scanf("%d",&c[i].b[3].capacity);

SetColor(24);

printf("Your college is registered successfully\n\n");

totcol++;//increasing total no.of colleges.

fp=fopen("college.txt","a+");

fprintf(fp,"%s %d %s %d %s %d %s %d %s %d\n",c[i].collname,c[i].rgno,c[i].b[0].name,c[i].b[0].capacity,c[i].b[1].name,c[i].b[1].capacity,c[i].b[2].name,c[i].b[2].capacity,c[i].b[3].name,c[i].b[3].capacity);

fclose(fp);

break;

case 2:i=totstud;

SetColor(15);

printf("\nEnter your Roll NO.\n");

scanf("%d",&s[i].roll);

SetColor(18);

printf("Enter your Total Marks\n");

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

SetColor(19);

printf("Enter your Name\n");

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

SetColor(22);

printf("Enter college choice 1\n");

scanf("%s",s[i].p[0].cname);

SetColor(23);

printf("Enter branch 1\n");

scanf("%s",s[i].p[0].cbranch);

SetColor(24);

printf("Enter college choice 2\n");

scanf("%s",s[i].p[1].cname);

SetColor(25);

printf("Enter branch 2\n");

scanf("%s",s[i].p[1].cbranch);

SetColor(26);

printf("Enter college choice 3\n");

scanf("%s",s[i].p[2].cname);

SetColor(27);

printf("Enter Branch 3\n");

scanf("%s",s[i].p[2].cbranch);

SetColor(28);

printf("Successfully Registered\n");

SetColor(15);

totstud++;

fp=fopen("student.txt","a+");

fprintf(fp,"%d %d %s %s %s %s %s %s %s\n",s[i].roll,s[i].marks,s[i].names,s[i].p[0].cname,s[i].p[0].cbranch,s[i].p[1].cname,s[i].p[1].cbranch,s[i].p[2].cname,s[i].p[2].cbranch);

fclose(fp);

break;

case 3:x=1;

break;

}

if(x!=1)

{

SetColor(30);

printf("Press 1 to end the registration process or any no. to continue registration\n");

scanf("%d",&x);

}

}

SetColor(28);

printf("\t\t\t\tList of colleges along with their no. of seats in different branches are as follows\n\n");

SetColor(15);

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

{

SetColor(18+i);

printf("\t\t\t\t\t\t%s %d %s %d %s %d %s %d %s %d\n",c[i].collname,c[i].rgno,c[i].b[0].name,c[i].b[0].capacity,c[i].b[1].name,c[i].b[1].capacity,c[i].b[2].name,c[i].b[2].capacity,c[i].b[3].name,c[i].b[3].capacity);

printf("\n");

}

SetColor(15);

sort(s,0,totstud);

for(i=0;i<totstud;i++) //i-selecting students one by one

{

flag=0;

for(j=0;j<totcol;j++) //j-searching college name one by one in the college list.

{

x=strcmp(s[i].p[0].cname,c[j].collname);

if(!x)

break;

}

for(k=0;k<4;k++) //k-searching opted branch by the student

{

y=strcmp(c[j].b[k].name , s[i].p[0].cbranch);

if(!y && (c[j].b[k].capacity>0))

{

flag=1;

strcpy(s[i].q.cname,c[j].collname);

strcpy(s[i].q.cbranch,c[j].b[k].name);

(c[j].b[k].capacity)--;

break;

}

}

if(flag==0)

{

for(j=0;j<totcol;j++)

{

x=strcmp(s[i].p[1].cname , c[j].collname);

if(!x)

break;

}

for(k=0;k<4;k++)

{

y=strcmp(c[j].b[k].name , s[i].p[1].cbranch);

if(!y && (c[j].b[k].capacity>0))

{

flag=1;

strcpy(s[i].q.cname,c[j].collname);

strcpy(s[i].q.cbranch,c[j].b[k].name);

(c[j].b[k].capacity)--;

break;

}

}

}

if(flag==0)

{

for(j=0;j<totcol;j++)

{

x=strcmp(s[i].p[2].cname , c[j].collname);

if(!x)

break;

}

for(k=0;k<4;k++)

{

y=strcmp(c[j].b[k].name , s[i].p[2].cbranch);

if(!y && (c[j].b[k].capacity>0))

{

flag=1;

strcpy(s[i].q.cname,c[j].collname);

strcpy(s[i].q.cbranch,c[j].b[k].name);

(c[j].b[k].capacity)--;

break;

}

}

}

if(flag==0)

{

strcpy(s[i].q.cname,"N.A");

strcpy(s[i].q.cbranch,"N.A");

}

}

fp=fopen("final.txt","w");

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

{

fprintf(fp,"%d %s %s %s\n",s[i].roll,s[i].names,s[i].q.cname,s[i].q.cbranch);

}

fprintf(fp,"\n\n\n\n");

fclose(fp);

SetColor(36);

printf("If you want Final Result to be displayed on terminal then press 1\n");

SetColor(39);

printf("If not then 0\n");

scanf("%d",&x);

SetColor(15);

if(x==1)

{

SetColor(19);

printf("\t\t\t\t\t\tFINAL RESULT IS AS FOLLOWS\n\n");

SetColor(15);

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

{

printf("\t\t\t\t\t\t%d %s %s %s\n",s[i].roll,s[i].names,s[i].q.cname,s[i].q.cbranch);

}

SetColor(23);

printf("N.A - Not Alloted\n");

}

printf("\n\n");

x=0;

char rope[20];

struct student temp1;

while(x!=1)

{

SetColor(29);

printf("\n\n\*\*1.See the available seats in a particular college\n");

SetColor(34);

printf("\*\*2.Search for a particular student\n");

SetColor(37);

printf("\*\*3.Allot college to a new student\n\n");

SetColor(42);

printf("\*\*4.To Exit\n");

SetColor(15);

printf("Enter choice\n");

scanf("%d",&ch);

switch(ch)

{

SetColor(16);

case 1:printf("Enter the college name you want to search\n");

scanf("%s",rope);

flag=0;

SetColor(15);

for(i=0;i<totcol;i++) // TOTal\_COLlege colleges are taken

{

x=strcmp(rope,c[i].collname);

if(!x)

{

printf("\n\n");

printf("\t\t\t\tName: %s Reg.No: %d %s: %d %s: %d %s: %d %s: %d\n\n",c[i].collname,c[i].rgno,c[i].b[0].name,c[i].b[0].capacity,c[i].b[1].name,c[i].b[1].capacity,c[i].b[2].name,c[i].b[2].capacity,c[i].b[3].name,c[i].b[3].capacity);

flag=1;

break;

}

}

if(flag==0)

printf("\n\nSorry, No such college exist.\nPlease check the spelling and try again.\n\n");

SetColor(9);

break;

case 2: SetColor(19);

printf("Enter the Roll No. of the student you want to search\n");

SetColor(11);

scanf("%d",&temp);

int t=binarysearch(s,0,totstud,temp);

SetColor(29);

if(t==-1)

printf("\t\t\t\tStudent not found with this roll no.\n");

else

printf("\n\t\t\t\tRoll No: %d Name: %s College: %s Branch: %s\n",s[t].roll,s[t].names,s[t].q.cname,s[t].q.cbranch);

break;

case 3: SetColor(26);

printf("Enter Your Roll. No.\n");

scanf("%d",&temp1.roll);

SetColor(27);

printf("Enter Your Total Score\n");

scanf("%d",&temp1.marks);

SetColor(54);

printf("Enter Your Name\n");

scanf("%s",temp1.names);

SetColor(57);

printf("Now give your preferences\n");

SetColor(44);

printf("College 1 and branch you want to opt for in it\n");

scanf("%s %s",temp1.p[0].cname,temp1.p[0].cbranch);

SetColor(45);

printf("College 2 and branch you want to opt for in it\n");

scanf("%s %s",temp1.p[1].cname,temp1.p[1].cbranch);

SetColor(51);

printf("College 3 and branch you want to opt for in it\n");

scanf("%s %s",temp1.p[2].cname,temp1.p[2].cbranch);

SetColor(15);

totstud++;

fp=fopen("student.txt","a+");

fprintf(fp,"\n%d %d %s %s %s %s %s %s %s",temp1.roll,temp1.marks,temp1.names,temp1.p[0].cname,temp1.p[0].cbranch,temp1.p[1].cname,temp1.p[1].cbranch,temp1.p[2].cname,temp1.p[2].cbranch);

fclose(fp);

printf("\n\nYour Data Has Been Saved Successfully\n\n");

flag=0;

for(j=0;j<totcol;j++) //j-searching college name one by one in the college list.

{

x=strcmp(temp1.p[0].cname,c[j].collname);

if(!x)

break;

}

for(k=0;k<4;k++) //k-searching opted branch by the student

{

y=strcmp(c[j].b[k].name , temp1.p[0].cbranch);

if(!y && (c[j].b[k].capacity>0))

{

flag=1;

strcpy(temp1.q.cname,c[j].collname);

strcpy(temp1.q.cbranch,c[j].b[k].name);

(c[j].b[k].capacity)--;

break;

}

}

if(flag==0)

{

for(j=0;j<totcol;j++)

{

x=strcmp(temp1.p[1].cname , c[j].collname);

if(!x)

break;

}

for(k=0;k<4;k++)

{

y=strcmp(c[j].b[k].name , temp1.p[1].cbranch);

if(!y && (c[j].b[k].capacity>0))

{

flag=1;

strcpy(temp1.q.cname,c[j].collname);

strcpy(temp1.q.cbranch,c[j].b[k].name);

(c[j].b[k].capacity)--;

break;

}

}

}

if(flag==0)

{

for(j=0;j<totcol;j++)

{

x=strcmp(temp1.p[2].cname , c[j].collname);

if(!x)

break;

}

for(k=0;k<4;k++)

{

y=strcmp(c[j].b[k].name , temp1.p[2].cbranch);

if(!y && (c[j].b[k].capacity>0))

{

flag=1;

strcpy(temp1.q.cname,c[j].collname);

strcpy(temp1.q.cbranch,c[j].b[k].name);

(c[j].b[k].capacity)--;

break;

}

}

}

if(flag==0)

{

//printf("flag raised\n");

strcpy(temp1.q.cname,"N.A");

strcpy(temp1.q.cbranch,"N.A");

}

fp=fopen("final.txt","a+");

fprintf(fp,"\n%d %s %s %s\n",temp1.roll,temp1.names,temp1.q.cname,temp1.q.cbranch);

fclose(fp);

SetColor(36);

printf("If you Want Final Result to be displayed on terminal then press 1\n");

SetColor(39);

printf("If not then 0\n");

scanf("%d",&x);

SetColor(15);

if(x==1)

{

SetColor(19);

printf("\t\t\t\t\t\tYour Result Is\n\n");

SetColor(15);

printf("\t\t\t\t%d %s %s %s\n",temp1.roll,temp1.names,temp1.q.cname,temp1.q.cbranch);

SetColor(23);

printf("\nN.A - Not Alloted\n");

}

printf("\n\n");

break;

case 4: exit(0);

break;

}

printf("Enter 1 to exit\n");

printf("OR any key to continue\n");

scanf("%d",&x);

}

}