#include <iostream>

#include <string.h>

using namespace std;

struct student

{

int roll;

char name[20];

float marks;

};

void swap(student &a, student &b)

{

student temp = a;

a = b;

b = temp;

}

void create(student s[], int n)

{

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

{

cout << "Enter roll no:" ;

cin >> s[i].roll;

cout << "Enter name:";

cin >> s[i].name;

cout << "Enter marks:" ;

cin >> s[i].marks;

}

}

void display(student s[], int n)

{

cout << "roll no "

<< " "

<< "name "

<< " "

<< "marks " << endl;

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

{

cout << s[i].roll << "\t" << s[i].name << "\t" << s[i].marks << endl;

}

}

void bubble(student s[], int n)

{

int i, j;

student temp;

for (i = 1; i < n; i++)

{

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

{

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

{

temp = s[j];

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

s[j + 1] = temp;

}

}

}

}

void insertion(student s[], int n)

{

for (int i = 1; i < n; i++)

{

student key;

key = s[i];

int j = i - 1;

while (j >= 0 && strcmp(s[j].name, key.name) > 0)

{

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

j--;

}

s[j + 1] = key;

}

}

int partition(student s[], int l, int u)

{

u--;

int p, q;

student pi = s[l];

p = l;

q = u;

while (p < q)

{

while (pi.marks <= s[p].marks)

{

p++;

}

while (pi.marks > s[q].marks)

{

q--;

}

if (p < q)

{

swap(s[p], s[q]);

}

}

swap(s[l], s[q]);

return q;

}

void quick(student s[], int l, int u)

{

if (l < u)

{

int pi = partition(s, l, u);

quick(s, l, pi - 1);

quick(s, pi + 1, u);

}

}

void binsearch(student s[], int l, int u)

{

int mid;

insertion(s, u);

cout << "Enter name to search" << endl;

char x[20];

cin >> x;

while (l <= u)

{

mid = (l + u) / 2;

if (strcmp(s[mid].name, x) == 0)

{

cout << "Student found!!" << endl;

cout << s[mid].roll << "\t" << s[mid].name << "\t" << s[mid].marks << endl;

return;

}

else if (strcmp(s[mid].name, x) < 0)

{

l = mid + 1;

}

else

{

u = mid - 1;

}

}

cout << "Not found!!" << endl;

}

void same\_sgpa(student s[],int n)

{

float marks;

cout<<"\n Enter the marks of student ";

cin>>marks;

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

{

if(marks==s[i].marks)

{

cout << s[i].roll << "\t" << s[i].name << "\t" << s[i].marks << endl;

}

}

}

int main()

{

student s[20];

int ch = 1,n;

do

{

cout << "1 .Create record:" << endl;

cout << "2 .Bubble sort :" << endl;

cout << "3 .Insertion sort :" << endl;

cout << "4 .Quicksort :" << endl;

cout<< "5 .Student with same SGPA :" <<endl;

cout << "6 .Binary search :" << endl;

cout << "\n Enter ur choice :" ;

cin >> ch;

switch (ch)

{

case 1:

cout << "Enter no of students\n";

cin >> n;

create(s, n);

display(s, n);

break;

case 2:

bubble(s, n);

display(s, n);

break;

case 3:

insertion(s, n);

display(s, n);

break;

case 4:

quick(s, 0, n);

display(s, n);

break;

case 5:

same\_sgpa(s,n);

break;

case 6:

binsearch(s, 0, n);

break;

}

cout<<"\n Do u want to continue";

cin>>ch;

}while(ch!=7);

return 0;

}

#include <iostream>

#include <string.h>

using namespace std;

struct student

{

int roll;

char name[20];

float marks;

};

void swap(student &a, student &b)

{

student temp = a;

a = b;

b = temp;

}

void create(student s[], int n)

{

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

{

cout << "Enter roll no:" ;

cin >> s[i].roll;

cout << "Enter name:";

cin >> s[i].name;

cout << "Enter marks:" ;

cin >> s[i].marks;

}

}

void display(student s[], int n)

{

cout << "roll no "

<< " "

<< "name "

<< " "

<< "marks " << endl;

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

{

cout << s[i].roll << "\t" << s[i].name << "\t" << s[i].marks << endl;

}

}

void bubble(student s[], int n)

{

int i, j;

student temp;

for (i = 1; i < n; i++)

{

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

{

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

{

temp = s[j];

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

s[j + 1] = temp;

}

}

}

}

void insertion(student s[], int n)

{

for (int i = 1; i < n; i++)

{

student key;

key = s[i];

int j = i - 1;

while (j >= 0 && strcmp(s[j].name, key.name) > 0)

{

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

j--;

}

s[j + 1] = key;

}

}

int partition(student s[], int l, int u)

{

u--;

int p, q;

student pi = s[l];

p = l;

q = u;

while (p < q)

{

while (pi.marks <= s[p].marks)

{

p++;

}

while (pi.marks > s[q].marks)

{

q--;

}

if (p < q)

{

swap(s[p], s[q]);

}

}

swap(s[l], s[q]);

return q;

}

void quick(student s[], int l, int u)

{

if (l < u)

{

int pi = partition(s, l, u);

quick(s, l, pi - 1);

quick(s, pi + 1, u);

}

}

void binsearch(student s[], int l, int u)

{

int mid;

insertion(s, u);

cout << "Enter name to search" << endl;

char x[20];

cin >> x;

while (l <= u)

{

mid = (l + u) / 2;

if (strcmp(s[mid].name, x) == 0)

{

cout << "Student found!!" << endl;

cout << s[mid].roll << "\t" << s[mid].name << "\t" << s[mid].marks << endl;

return;

}

else if (strcmp(s[mid].name, x) < 0)

{

l = mid + 1;

}

else

{

u = mid - 1;

}

}

cout << "Not found!!" << endl;

}

void same\_sgpa(student s[],int n)

{

float marks;

cout<<"\n Enter the marks of student ";

cin>>marks;

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

{

if(marks==s[i].marks)

{

cout << s[i].roll << "\t" << s[i].name << "\t" << s[i].marks << endl;

}

}

}

int main()

{

student s[20];

int ch = 1,n;

do

{

cout << "1 .Create record:" << endl;

cout << "2 .Bubble sort :" << endl;

cout << "3 .Insertion sort :" << endl;

cout << "4 .Quicksort :" << endl;

cout<< "5 .Student with same SGPA :" <<endl;

cout << "6 .Binary search :" << endl;

cout << "\n Enter ur choice :" ;

cin >> ch;

switch (ch)

{

case 1:

cout << "Enter no of students\n";

cin >> n;

create(s, n);

display(s, n);

break;

case 2:

bubble(s, n);

display(s, n);

break;

case 3:

insertion(s, n);

display(s, n);

break;

case 4:

quick(s, 0, n);

display(s, n);

break;

case 5:

same\_sgpa(s,n);

break;

case 6:

binsearch(s, 0, n);

break;

}

cout<<"\n Do u want to continue";

cin>>ch;

}while(ch!=7);

return 0;

}