**Assignment No :1**

Name: Rasika Mahure

PRN:72018258B

Div:B

Roll No: S1951093

**Aim :-**

Consider a student database of SEIT class (at least 15 records). Database contains different fields of every student like Roll No, Name and SGPA.(array of structure)

1. Design a roll call list, arrange list of students according to roll numbers in ascending order (Use

Bubble Sort)

1. Arrange list of students alphabetically. (Use Insertion sort)
2. Arrange list of students to find out first ten toppers from a class. (Use Quick sort)
3. Search students according to SGPA. If more than one student having same SGPA, then print list of all students having same SGPA.

Search a particular student according to name using binary search without recursion. (all the student records having the presence of search key should be displayed)

# //Program

#include <iostream> #include <string.h> using namespace std; typedef struct student { int roll\_num; char name[20]; float marks;

} stud; void create(stud s[20], int n); void display(stud s[20], int n); void bubble\_sort(stud s[20], int n); void insertionSort(stud s[20], int n); void quick\_sort(stud s[20], int, int); int partition(stud s[20], int, int); void search(stud s[20], int n, int key); int bsearch(stud s[20], char x[20], int low, in t high); int main() { stud s[20]; int ch, n, key, result; char x[20]; do

{ cout << "\n 1) Create Student Database

"; cout << "\n 2) Display Student

Records

"; cout << "\n 3) Bubble Sort "; cout << "\n 4) Insertion Sort "; cout << "\n 5) Quick Sort "; cout << "\n 6) Linear search "; cout

<< "\n 7) Binary search "; cout

<< "\n 8) Exit "; cout << "\n Enetr Your Choice:="; cin >> ch; switch (ch) { case 1:

cout << "\n Enter The Number Of Rec ords:="; cin >> n; create(s, n);

break; case 2:

display(s, n); break; case 3: bubble\_sort(s, n); break; case 4: insertionSort(s, n); break; case 5:

quick\_sort(s, 0, n - 1); cout << "\n" <<

"\t"

<< "Roll No"

<< "\t"

<< " Name"

<< "\t"

<< "Marks"; for (int i = n - 1; i >= n - 10; i-

-) { cout << "\n"; cout << "\t " << s[i].roll\_num

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

} break; case 6: cout << "\n Enter the marks which u want to search:="; cin >> key; search(s, n, key); break; case 7:

cout << "\n Enter the name of stude nt which u want to search:="; cin

>> x;

insertionSort(s, n); result

= bsearch(s, x, 0, (n - 1)); if (result == 1)

{ cout << " \n Student name you w

ant to search for is not present ! \n";

} else { cout << " \n The student is pr esent :\t" << s[result].name; }

break; case 8:

return 0; default:

cout << "\n Invalid choice !! Pleas e enter your choice again." << endl;

}

} while (ch != 8);

} void create(stud s[20], int n)

{ int i; for (i =

0; i < n; i++)

{ cout << "\n Enter the roll number:="; cin >> s[i].roll\_num; cout << "\n Enter the Name:="; cin >> s[i].name; cout << "\n Enter the marks:="; cin >> s[i].marks;

} } void display(stud s[20], int n)

{ int i; cout << "\n"

<< "\t"

<< "Roll No"

<< "\t"

<< " Name"

<< "\t" << "Marks"; for (i = 0; i

< n; i++)

{

cout << "\n"; cout << "\t " << s[i].roll\_num << "\t "

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

}

}

//bubble sort to sort in ascending order on rol l number void bubble\_sort(stud s[20], int n)

{ int i, j; stud temp; for (i = 1; i < n; i++)

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

{ if (s[j].roll\_num > s[j + 1].roll\_n um) { temp = s[j]; s[j] = s[j + 1]; s[j + 1] = temp;

}

}

}

}

// insertion sort to sort on names in ascending order void insertionSort(stud s[20], int n)

{ int i, j; stud key; for (i = 1; i < n; i++) { key = s[i]; j = i

- 1;

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

name) > 0)

{

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

j = j - 1;

} s[j + 1] = key;

}

}

//Quick sort to sort on marks void quick\_sort(stud s[20], int l, int u)

{ int j; if (l < u)

{ j = partition(s, l,

u); quick\_sort(s, l, j - 1); quick\_sort(s, j + 1, u); }

} int partition(stud s[20], int l, int u)

{ int i, j; stud temp, v; v = s[l]; i = l; j = u + 1; do { do i++; while (s[i].marks

< v.marks && i <= u); do j--; while (v.marks < s[j].marks); if (i < j) { temp = s[i]; s[i] = s[j]; s[j] = temp;

}

} while (i < j); s[l] = s[j]; s[j] = v; return (j);

}

// linear search for marks if more than one stu dent having same marks print all of them void search(stud s[20], int n, int key) { int i; cout << "\n"

<< "\t"

<< "Roll No"

<< "\t"

<< " Name"

<< "\t" << "Marks"; for (i = 0; i < n; i++)

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

{ cout << "\n\t " << s[i].roll\_num <<

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

}

} } int bsearch(stud s[20], char x[20], int low, in t high) { int mid; while (low <= high) { mid = (low + high) / 2; if (strcmp(x, s[mid].name) == 0)

{

return mid;

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

< 0)

{ high = mid - 1;

} else

{ low = mid +

1;

} } return -

1;

}

**OUTPUT**

1. Create Student Database
2. Display Student Records
3. Bubble Sort
4. Insertion Sort 5) Quick Sort
5. Linear search
6. Binary search
7. Exit Enter Your choice:= 1

Enter the number of records:=

15

Enter the roll number:= 15

Enter the Name:= Rasika

Enter the marks:= 44

Enter the roll number:= 1

Enter the Name:= Palak

Enter the marks:= 34

Enter the roll number:= 6

Enter the Name:= John

Enter the marks:= 32

Enter the roll number:= 8

Enter the Name:= Elon

Enter the marks:= 20

Enter the roll number:= 3

Enter the Name:= Johny

Enter the marks:= 45

Enter the roll number:= 2

Enter the Name:= Jimmy

Enter the marks:= 34

Enter the roll number:= 4 Enter the Name:= Rachit

Enter the marks:= 37

Enter the roll number:= 9

Enter the Name:= Vaibhavi

Enter the marks:= 43

Enter the roll number:= 10

Enter the Name:= Vaibhav

Enter the marks:= 35

Enter the roll number:= 14

Enter the Name:= Akshata

Enter the marks:= 38

Enter the roll number:= 12

Enter the Name:= Gajanan

Enter the marks:= 33

Enter the roll number:= 5 Enter the Name:= Saniya

Enter the marks:= 33

Enter the roll number:= 7

Enter the Name:= Akshit Enter the marks:= 39

Enter the roll number:= 11

Enter the Name:= Ishika

Enter the marks:= 41

Enter the roll number:= 13

Enter the Name:= Lisha

Enter the marks:= 40

1. Create Student Database
2. Display Student Records
3. Bubble Sort
4. Insertion Sort
5. Quick Sort
6. Linear search 7) Binary search

8) Exit Enter Your choice:= 2

Roll No Name Marks

15 Rasika 44

1 Palak 34

6 John 32

8 Elon 20

1. Johny 45

2 Jimmy 34

1. Rachit 37
2. Vaibhavi 43
3. Vaibhav 35 14 Akshata 38

12 Gajanan 33

5 Saniya 33

7 Akshit 39

11 Ishika 41

13 Lisha 40

1. Create Student Database
2. Display Student Records
3. Bubble Sort
4. Insertion Sort
5. Quick Sort
6. Linear search 7) Binary search

8) Exit Enter Your choice:= 3

1. Create Student Database
2. Display Student Records
3. Bubble Sort
4. Insertion Sort
5. Quick Sort
6. Linear search 7) Binary search

8) Exit Enter Your choice:= 2

Roll No Name Marks 1 Palak 34

1. Jimmy 34
2. Johny 45 4 Rachit 37
3. Saniya 33
4. John 32
5. Akshit 39
6. Elon 20
7. Vaibhavi 43
8. Vaibhav 35
9. Ishika 41
10. Gajanan 33
11. Lisha 40
12. Akshata 38
13. Rasika 44
14. Create Student Database
15. Display Student Records
16. Bubble Sort
17. Insertion Sort
18. Quick Sort
19. Linear search 7) Binary search

8) Exit Enter Your choice:= 4

1) Create Student Database 2)

Display Student Records

1. Bubble Sort
2. Insertion Sort
3. Quick Sort
4. Linear search 7) Binary search

8) Exit Enter Your choice:= 2

Roll No Name Marks

14 Akshata 38

1. Akshit 39
2. Elon 20

12 Gajanan 33

11 Ishika 41

1. Jimmy 34

6 John 32

1. Johny 45 13 Lisha 40

1 Palak 34

1. Rachit 37

15 Rasika 44

1. Saniya 33

10 Vaibhav 35

9 Vaibhavi 43

1. Create Student Database
2. Display Student Records
3. Bubble Sort
4. Insertion Sort
5. Quick Sort
6. Linear search 7) Binary search

8) Exit Enter Your choice:= 5 Roll No Name Marks 3 Johny 45

15 Rasika 44

9 Vaibhavi 43

11 Ishika 41

1. Lisha 40 7

Akshit 39

1. Akshata 38

4 Rachit 37

10 Vaibhav 35

1 Palak 34

1. Create Student Database
2. Display Student Records
3. Bubble Sort
4. Insertion Sort
5. Quick Sort
6. Linear search
7. Binary search
8. Exit Enter Your choice:= 6

Enter the marks which u want to search:=44

Roll No Name Marks

15 Rasika 44

1. Create Student Database
2. Display Student Records
3. Bubble Sort
4. Insertion Sort
5. Quick Sort
6. Linear search
7. Binary search
8. Exit Enter Your choice:= 7

Enter the name of student which u want to search:=Akshata

The student is present : Akshata

1. Create Student Database
2. Display Student Records
3. Bubble Sort
4. Insertion Sort
5. Quick Sort
6. Linear search
7. Binary search
8. Exit Enter Your choice:=