createIndexFile()

In this program index file is used to store the student numbers(as indices) and their offset(address) that are gotten from the record file.

Indexing helps to locate the position of a specific record in a record file in a short period of time. It helps to make a quick decision by providing necessary information stored in files.

for instance, when a specific record needs to be searched, updated, or deleted searching it on an index file and applying the changes saves time that would have otherwise been wasted by going through a bunch of data in record files. To create and store records on an index file: first, a structure of type 'ogrenci' is declared and initialized dynamically. after that the record file and index file opened in read(rb) and write(w) mode respectively. Student number and offset of each record are read from the record file using fread() function and written on the index file using fprintf() function in a while loop. Then the data in the index files are transferred to a dynamically initialized array to be sorted using a bubble sort algorithm. at the end, the sorted indices with their offsets are written back to the index file using fprintf() function in a for loop and both index and record files closed.

findRecord()

The findRecord() function returns a record at a specified index from the record file where the key value(which is student number) matches the specified predicate.

The function prints the message "record is not found" if no record is found for the given key(student number). To do this first the function gets student number from the user and it searches for the specified student number in the index file using modified binary search. The binary search is modified to be used in conditions where there are multiple records with the same key(student number). If the specified student number is found in the index file, the function moves the file pointer of the record file to a searched record's location using the offset information gotten from the index file with fseek() function. Then the record is read from the record file using fread() function. Lastly, the function displays the searched record in a formatted way.

Displays all records in the datafile when the ViewDatafile() function is called.

To do this, a file pointer and a struct of type 'student' are declared. Then the data file is opened in binary read mode (rb). Then using the fread() function as a condition in a while loop, the function prints each record in the data file using printf in the loop. fread is used in a while loop because calling it while(fread(&s1, sizeof(student), 1, file)) advances the file pointer by the student size of the struct each time.