FINAL PROJECT REPORT

BIC10404 : DATA STRUCTURE

TITLE	STUDENT HOME SPORT CHECKING SYSTEM	
GROUP NAME	AESTA FIGO	
SECTION	11 & 12	
GROUP MEMBERS' NAME		**MATRIC NO.**
	MOHAMAD ARIF AZINUDDIN BIN ZAIDI **(LEADER)**	AI210125
	MUHAMMAD AIMAN BIN MOHD LATIFF	AI210258
	MUHAMMAD AIMAN SHAHZRIN BIN MOHD OTHMAN	AI210066
	ZALIFF BIN USRI	AI210252
	MOHAMAD IRFAN BIN ABDUL MUTHALIF	AI210249
LECTURER'S NAME	PROFESSOR MADYA DR. NOOR AZAH BINTI SAMSUDIN	

TABLE OF CONTENTS

ACKNOWLEDGEMENT

First of all, we would like to thank God for enabling us to finally finish the task given to us by the instructor on data structure. While working on this task, the group dealt with a few small concerns to the best of their abilities. Thankfully, all of the worries have been put to rest, and we have been able to sensibly and effectively react.

We would especially like to thank Prof. Madya Dr. Noor Azah Binti Samsudin, our professor, without whose help our project would not have been as successful as it was. She is constantly ready to assist and give us advice on how to carry out these undertakings. She gave us excellent direction and stood with us when things became tough. Her involvement was essential to the project's successful completion.

Finally, many thanks to our great friends who have always worked hard to deliver a wonderful assignment with all affordance and responsibility. We hope that all of our efforts will bear fruit for us and our group initiative. We also want to thank all of our classmates for helping us with our group project. They are always offering us recommendations and comments on our projects, helping us to improve them in a number of ways. We would like to thank everyone who helped and partnered with us to make this effort a success.

CHAPTER 1 : INTRODUCTION

1.1 Project Title

For the introduction, from the knowledge that we have learned from the course BIC 10404 Data structure, we have build or created a system which is very dependable and useful mainly for the school management which was handle by teachers. Besides that, after we do discussion within our fellow group members, we come up with an idea creating a Student Home Sport Checking System for any kinds of school such as primary and secondary school both can use this system. It is because, When its comes to any sport events or activities that will be organized by the school there is always student that gets their home sport mixed when the events ongoing and change their home sport without informing or asking teacher for the permission. Its makes the teachers hard to manage the sports events because the large home sports members always gets the title winner because of their quantity is bigger than other home sports members. But, now we already make a system that will help the teachers to put all the students information about their home sport together with their ID number, class, name and sport events which that the teacher can search the student using either of the data.

1.2 Project Description

Despite of that, with the knowledge of the data structure course that we have learned, we have build this project using c language and implement the coding of the system in Dev-C++ IDE. We have put some order collection of elements such as array, function, searching, sorting and struct that we studied from this course in our project for make it more easy to use and get what we needed. Firstly, we have coded the use of an array of structure for store the data of students marks. Our programming includes 8 functions including the main function. For the starting, we created a main function for the declaration of the variables of the information that needed by the system. Next function, we put a pop-up instruction such as Input, Display, Search, Sorted, Delete, Update and to end program according the teachers of the system. Third, we inserted function about asking input from the teachers that needed by the system such as Name, class, home sport and sport event . The fourth function is to display the data about all the students that stored in the system or the certain student that needed by teachers. The fifth and sixth function is for searching and sorting the data that

already stored for easy to find and sorts them by their attributes in the data according the user. Besides that, the seventh function is, update function which is needed by all kind of the system is where the teachers can change their data about a students if the students wanted to change their home sports or any data that related in the system. Thus, we added deletion function that the teachers can used if any of the students shift their school to other place and no need their data anymore.

CHAPTER 2 : CODING

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <ctype.h>
#define BUFFER_SIZE 4096

//make struct Information
struct Information
{
    char Name[20];
    char Class[20];
    int Id;
    char HomeSports[20];
    char SportEvent[20];


};

//to change name of the struct
typedef struct Information Information;

//struct Node
struct Node
{
    //declaration of data with datatype Information
    Information data;
    struct Node *nextPtr;
};

```

```
31  typedef struct Node Node;
32  Node *headPtr=NULL,
33       *newPtr,
34       *prevPtr,
35       *currPtr,
36       *tempPtr;
37
38  Information inputData;
39
40  //declaration of function
41  bool parse_int(char *string,int *Id);
42  void display();
43  int search(int Id);
44  void update();
45  void deleted();
46  void sort();
47  void Sorted();
48  void Instruction();
49  void menu(Information data);
50
51  int main()
52  {
53      char Name[20];
54      char Class[20];
55      int Id;
56      char HomeSports[20];
57      char SportEvent[20];
58
59      headPtr = NULL;
60      int value;
```

```
62      //switch case use for user to enter what they want to do
63      while(value!=7)
64      {
65          Instruction();
66          printf("\n\tEnter your choice:");
67          scanf("%d",&value);
68          switch(value)
69          {
70              case 1://for user to input data
71                  system("cls");
72                  menu(inputData);
73                  break;
74
75              case 2://for system to display data
76                  system("cls");
77                  display();
78                  break;
79
80              case 3://for system to search data
81                  system("cls");
82                  displayS();
83                  break;
84
85              case 4://for system to sort data
86                  system("cls");
87                  Sorted();
88                  break;
89
90              case 5://for system to delete data
91                  system("cls");
92                  deleted();
93                  break;
94
95              case 6://for system to update data
96                  system("cls");
97                  update();
98                  break;
99
100
101
102         }
103     }
104     printf("End Program");
105 }
```

```c
//function instruction
void Instruction()
{

    printf("\n\tYour Choice:");
    printf("\n\t ################");
    printf("\n\t #1.Input       #");
    printf("\n\t #2.Display     #");
    printf("\n\t #3.Search      #");
    printf("\n\t #4.Sorted      #");
    printf("\n\t #5.Delete      #");
    printf("\n\t #6.Update      #");
    printf("\n\t #7.End Program #");
    printf("\n\t ################");
}

//function to check data is integer or not
bool parse_int(char *string,int *Id)
{
    int i=0;

    while (isspace(string[i])) i++;

    // strlen will return the length of the string
    int length = strlen(string);

    if (length == i) return false;

    char integer_buffer[BUFFER_SIZE];
    int integer_chars = 0;


    while (i < length && !isspace(string[i]))//keep on looping one character at a time
    {
        if (!isdigit (string[i]))return false;
        integer_buffer[integer_chars] = string[i];

        integer_chars++;
        i++;
    }

    integer_buffer[integer_chars] = '\0';
```

```c
    while (isspace(string[i])) i++;//loop through

    if ( string[i] != '\0') return false;

    *Id = atoi(integer_buffer);//take the string representation integer function and make it actual integer value

    return true;
}

//function menu
void menu(Information data)
{


    int Id =0;
    bool parsed_correct =true;//declaration for checking data
    do
    {

        printf("\nPlease enter your id:");

        char buffer[BUFFER_SIZE];

        fgets(buffer, BUFFER_SIZE, stdin);//read from standard in and store the character into buffer character array up to buffer size

        parsed_correct = parse_int(buffer,&inputData.Id);//pass the integer pointer into integer variable

        if(!parsed_correct)
        {
            printf("Check again..");
        }
```

```
181
182     }while(!parsed_correct);
183
184     fflush (stdin);
185     printf("Please enter your Name:");
186     gets(inputData.Name);
187     printf("Please enter your class:");
188     gets(inputData.Class);
189     printf("Please enter your Home Sport:");
190     gets(inputData.HomeSports);
191     printf("Please enter your Sport Event:");
192     gets(inputData.SportEvent);
193     fflush (stdin);
194
195     //to allocate the memory of Node
196     newPtr=malloc(sizeof(Node));
```

```
199     if(newPtr !=NULL)
200     {
201         newPtr->data = inputData;//store data from input data
202         newPtr->nextPtr = NULL;
203
204         prevPtr=NULL;
205         currPtr=headPtr;//head pointer assign to currennt pointer
206
207         //loop to find the right position in the linked list
208         while (currPtr!=NULL && inputData.Id > currPtr->data.Id)
209         {
210             prevPtr = currPtr;
211             currPtr = currPtr->nextPtr;
212         }
213             //insert a new node at the beginning of a linked list
214             if (prevPtr == NULL)
215             {
216                 newPtr->nextPtr = headPtr;
217                 headPtr = newPtr;
218             }
219
220             else
221             {
222                 //insert a new node in the middle or at the end of the linked list
223                 prevPtr->nextPtr = newPtr;
224                 newPtr->nextPtr = currPtr;
225             }
226     }
227
228     else
229     printf("\nThe data are not inserted.Please insert back the data..");
230
231 }
```

```
233 //function display
234 void display()
235 {
236     //display content of the linked list
237     currPtr = headPtr;
238     if(headPtr == NULL)
239     {
240         printf("Ypur data is not found (>_<) Please try again.");
241 
242     }
243     else
244     {
245         while (currPtr !=NULL)
246         {
247             printf("\nId Number:%d",currPtr->data.Id);
248             printf("\nName:");
249             puts(currPtr->data.Name);
250             printf("Class:");
251             puts(currPtr->data.Class);
252             printf("Home Sport:");
253             puts(currPtr->data.HomeSports);
254             printf("Sport you entered:");
255             puts(currPtr->data.SportEvent);
256             currPtr = currPtr->nextPtr;
257         }
258     }
259 }
```

```
261 //function searching
262 int search(int searchingId)
263 {
264     int Search;
265 
266         //to check if the Id that the user search is the same as system
267         if(headPtr->data.Id == searchingId)
268         {
269             tempPtr = headPtr;
270             Search=1;
271         }
272         else
273         {
274             prevPtr = headPtr;
275             currPtr = headPtr ->nextPtr;
276 
277             //loop to find the right position in the linked list
278             while(currPtr!=NULL && currPtr->data.Id != searchingId)
279             {
280                 prevPtr = currPtr;
281                 currPtr = headPtr->nextPtr;
282             }
283 
284 
285             if( currPtr!=NULL)
286 
287             {
288                 tempPtr=currPtr;
289                 Search=2;
290             }
291             else
292             Search=-1;
293 
294         }
295     return Search;
296 }
```

```
298 int displayS()
299 {
300     int searching;
301     int location;
302 
303     printf("\n\n Please enter the data that you want to search:");
304     scanf("%d",&searching);
305 
306     //to call back the function search
307     location = search(searching);
308 
309     if(location != -1)
310     {
311         printf("\nId Number:%d",tempPtr->data.Id);
312         printf("\nName:");
313         puts(tempPtr->data.Name);
314         printf("Class:");
315         puts(tempPtr->data.Class);
316         printf("Home Sport:");
317         puts(tempPtr->data.HomeSports);
318         printf("Sport you entered:");
319         puts(tempPtr->data.SportEvent);
320     }
321     else
322     {
323         printf("The data that you search is not available in the store...\n");
324     }
325 
326 }
```

```c
//function deleted
void deleted()
{
    int deletion;
    int location1;

    //prompt user to enter data to be deleted
    printf("\n\nPlease enter the data that you want to delete:");
    scanf("%d",&deletion);

    //to call back function search
    location1 = search(deletion);


    while(location1 ==-1)
    {
        printf("Sorry!the data is not found.Please Try again!");
        scanf("%d",&deletion);

        location1 = search(deletion);
    }
    if(location1 == 1)
    {
        //delete the node search
        headPtr = headPtr->nextPtr;
        free(tempPtr);
    }
    else if(location1 ==2)
    {
        //delete the node search
        prevPtr->nextPtr = currPtr->nextPtr;
        free(tempPtr);

    }
    printf("Data deleted...");
    currPtr=headPtr;
}
```

```
366 //function sort
367 void sort()
368 {
369     int size=0;
370     Information hold;
371 
372     currPtr = headPtr;//to assign headPtr into currPtr
373 
374     //Keep track the list of node
375     while(currPtr !=NULL)
376     {
377         size =size+1;
378         currPtr= currPtr->nextPtr;
379     }
380 
381 
382     int i;
383 
384     //to make node sort in ascending order
385     for(i=1; i< size ; i++)
386     {
387         prevPtr = headPtr;
388         currPtr = prevPtr->nextPtr;
389 
390         int j;
391 
392         for(j=1 ; j<size ; j++)
393         {
394 
395             //to sort the Node in order
396             if(prevPtr->data.Id > currPtr->data.Id)
397             {
398                 hold = prevPtr->data;
399                 prevPtr->data= currPtr->data;
400                 currPtr->data = hold;
401             }
402 
403             prevPtr = currPtr;
404             currPtr = currPtr->nextPtr;
405         }
406     }
407 }
```

```c
//function Sorted
void Sorted()
{
    sort();//call function

    currPtr = headPtr;

    while(currPtr != NULL)
    {
        printf("\nId Number:%d",currPtr->data.Id);
        printf("\nName:",currPtr->data.Name);
        printf("Class:");
        puts(currPtr->data.Class);
        printf("Home Sport:");
        puts(currPtr->data.HomeSports);
        printf("Sport you entered:");
        puts(currPtr->data.SportEvent);

        currPtr = currPtr->nextPtr;
    }
}
```

```c
//update
void update()
{
    int updating;
    int location;
    int choice;


    //prompt user to enter which want to update
    printf("What do you want to update:");
    printf("\n1.Id");
    printf("\n2.Name");
    printf("\n3.Class");
    printf("\n4.HomeSport");
    printf("\n5.SportEvent");
```

```c
    //prompt user to enter their choice
    printf("\nPlease enter your choice:");
    scanf("%d",&choice);

    switch (choice)
    {


        case 1:

            //prompt user to enter data Id to update
            printf("\n\n Please enter the Id data that you want to update:");
            scanf("%d",&updating);


            location = search(updating);

            if(location != -1)
            {
                int Id;

```

```
469                 printf("\nSearching Id:%d",tempPtr->data.Id);
470                 printf("\nlocation founded!");
471                 fflush(stdin);
472                 printf("\nEnter New Id:");
473                 scanf("%d",&tempPtr->data.Id);
474                 printf("Your new data is :%d \n",tempPtr->data.Id);
475             }
476             else
477             {
478                 printf("The data that you search is not available in the store...");
479             }
480             break;
481 
482         case 2:
483 
484             //prompt user to enter data Id to update
485             printf("\n\n Please enter the data Id that you want to update:");
486             scanf("%d",&updating);
487 
488 
489 
490             location = search(updating);
491 
492             if(location != -1)
493             {
494                 int Id;
495 
496 
497                 printf("\nSearching Name for Id:%d",tempPtr->data.Id);
498                 printf("\nlocation founded!");
499                 fflush(stdin);
500                 printf("\n\nEnter New Name:");
501                 gets(tempPtr->data.Name);
502                 printf("Your new data is :");
503                 puts(tempPtr->data.Name);
504                 printf("\n");
505             }
506         else
507             {
508                 printf("The data that you search is not available in the store...");
509             }
510             break;
```

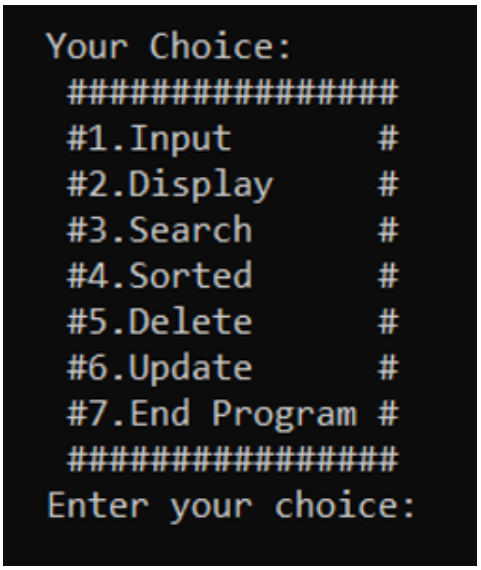
```
512         case 3:
513 
514             //prompt user to enter data Id to update
515             printf("\n\n Please enter the data Id that you want to update:");
516             scanf("%d",&updating);
517 
518 
519 
520             location = search(updating);
521 
522             if(location != -1)
523             {
524                 int Id;
525 
526 
527                 printf("\nSearching Class for Id:%d",tempPtr->data.Id);
528                 printf("\nlocation founded!");
529                 fflush(stdin);
530                 printf("\n\nEnter New Class:");
531                 gets(tempPtr->data.Class);
532                 printf("Your new data is :");
533                 puts(tempPtr->data.Class);
534                 printf("\n");
535             }
536             else
537             {
538                 printf("The data that you search is not available in the store...");
539             }
540             break;
541 
542         case 4:
543 
544             //prompt user to enter data Id to update
545             printf("\n\n Please enter the data Id that you want to update:");
546             scanf("%d",&updating);
```

```
550             location = search(updating);
551 
552             if(location != -1)
553             {
554                 int Id;
555 
556 
557                 printf("\nSearching Class for HomeSports:%d",tempPtr->data.Id);
558                 printf("\nlocation founded!");
559 
560                 fflush(stdin);
561                 printf("\nEnter New HomeSport:");
562                 gets(tempPtr->data.HomeSports);
563                 printf("Your new data is :");
564                 puts(tempPtr->data.HomeSports);
565                 printf("\n");
566             }
567             else
568             {
569                 printf("The data that you search is not available in the store...");
570             }
571             break;
572 
573         case 5:
574 
575             //prompt user to enter data Id to update
576             printf("\n\n Please enter the data Id that you want to update:");
577             scanf("%d",&updating);
```

```
581             location = search(updating);
582 
583             if(location != -1)
584             {
585                 int Id;
586 
587 
588                 printf("\nSearching Class for HomeSports:%d",tempPtr->data.Id);
589                 printf("\nlocation founded!");
590 
591                 fflush(stdin);
592                 printf("\nEnter New SportEvent:");
593                 gets(tempPtr->data.SportEvent);
594                 printf("Your new data is :");
595                 puts(tempPtr->data.SportEvent);
596                 printf("\n");
597             }
598             else
599             {
600                 printf("The data that you search is not available in the store...");
601             }
602             break;
603 
604 
605     }
606 
607 
608 
609 }
```

CHAPTER 3 : OUTPUT

When the programme is finished, complete, and can execute without any syntax or logic errors, it will display the seven options indicated below.



When the user selects option 1, the result will show the input that the user must enter. This includes the user id, name, class, home sport, and sport event. The main choice will then be displayed by the programme. To choose other alternatives, the user must submit input. The user's input will be saved in the database.

```
Please enter your id:Check again..
Please enter your id:1
Please enter your Name:AIMAN
Please enter your class:S12
Please enter your Home Sport:RED
Please enter your Sport Event:JUMPING

        Your Choice:
         ################
         #1.Input       #
         #2.Display     #
         #3.Search      #
         #4.Sorted      #
         #5.Delete      #
         #6.Update      #
         #7.End Program #
         ################
        Enter your choice:
```

If the user enters invalid data, the application will display the message 'Check again..' and the user must enter their information again.

```
Please enter your id:Check again..
Please enter your id:a12
Check again..
Please enter your id:s123
Check again..
Please enter your id:
```

When a user selects an option without entering their input or selects an invalid option, the application will show 'Your data is not found (>_<) Please try again.' and the main option will be displayed allowing the user to enter another option.

```
Ypur data is not found (>_<) Please try again.
        Your Choice:
         ################
         #1.Input       #
         #2.Display     #
         #3.Search      #
         #4.Sorted      #
         #5.Delete      #
         #6.Update      #
         #7.End Program #
         ################
        Enter your choice:
```

If user choice 2 the output will display all the data that user input.

```
Id Number:1
Name:AIMAN
Class:S12
Home Sport:RED
Sport you entered:JUMPING

Id Number:2
Name:SHAHZRIN
Class:S12
Home Sport:YELLOW
Sport you entered:RUNNING

Id Number:3
Name:MUHAMMAD
Class:S11
Home Sport:GREEN
Sport you entered:SWIMMING

        Your Choice:
         ################
         #1.Input       #
         #2.Display     #
         #3.Search      #
         #4.Sorted      #
         #5.Delete      #
         #6.Update      #
         #7.End Program #
         ################
        Enter your choice:
```

If the user selects option 3, the output will say, 'Please input the data that you wish to search for: ' and the output will present the data that the user need as well as the major options for the user to choose.

```
Please enter the data that you want to search:1

Id Number:1
Name:AIMAN
Class:S12
Home Sport:RED
Sport you entered:JUMPING

        Your Choice:
         ################
         #1.Input       #
         #2.Display     #
         #3.Search      #
         #4.Sorted      #
         #5.Delete      #
         #6.Update      #
         #7.End Program #
         ################
        Enter your choice:
```

If the user enters incorrect data, the result will show 'The data that you searched for is not accessible in the store...', and the main choice for the user to pick will be displayed.

```
Please enter the data that you want to search:The data that you search is not available in the store..

       Your Choice:
        ################
        #1.Input       #
        #2.Display     #
        #3.Search      #
        #4.Sorted      #
        #5.Delete      #
        #6.Update      #
        #7.End Program #
        ################
       Enter your choice:
```

When the user selects option 4, the output will show the sorted data that the user has saved, and the software will display the main option, from which the user may select the data that the user wants to obtain.

```
Id Number:1
Name:Class:S12
Home Sport:RED
Sport you entered:JUMPING

Id Number:2
Name:Class:S12
Home Sport:YELLOW
Sport you entered:RUNNING

Id Number:3
Name:Class:S11
Home Sport:GREEN
Sport you entered:SWIMMING

        Your Choice:
         ################
         #1.Input       #
         #2.Display     #
         #3.Search      #
         #4.Sorted      #
         #5.Delete      #
         #6.Update      #
         #7.End Program #
         ################
        Enter your choice:
```

When the user selects option 6, the output will display what the user want to edit, which is the id, name, class, and Homesport. The output will indicate the Id data that the user wants to update, as well as the searching id and the place discovered if the data is located in the data that has been stored. The output will display 'Enter New Id:' for the user and display new data that the user has updated and save it in the data that the user has updated, as well as display the main choice for the user to pick the next option.

```
What do you want to update:
1.Id
2.Name
3.Class
4.HomeSport
Please enter your choice:1


 Please enter the Id data that you want to update:1

Searching Id:1
location founded!
Enter New Id:4
Your new data is :4

        Your Choice:
         ################
         #1.Input       #
         #2.Display     #
         #3.Search      #
         #4.Sorted      #
         #5.Delete      #
         #6.Update      #
         #7.End Program #
         ################
        Enter your choice:
```

If the update choice is not incorrect, the software will present the main option for the user to select the option that the user requires.

```
What do you want to update:
1.Id
2.Name
3.Class
4.HomeSport
Please enter your choice:
        Your Choice:
         ################
         #1.Input       #
         #2.Display     #
         #3.Search      #
         #4.Sorted      #
         #5.Delete      #
         #6.Update      #
         #7.End Program #
         ################
        Enter your choice:
```

If the user selects option 7, the software will be terminated and the data will be erased.

```
What do you want to update:
1.Id
2.Name
3.Class
4.HomeSport
Please enter your choice:1


 Please enter the Id data that you want to update:1

Searching Id:1
location founded!
Enter New Id:4
Your new data is :4

        Your Choice:
         ################
         #1.Input       #
         #2.Display     #
         #3.Search      #
         #4.Sorted      #
         #5.Delete      #
         #6.Update      #
         #7.End Program #
         ################
        Enter your choice:7
End Program
--------------------------------
Process exited after 783.1 seconds with return value 0
Press any key to continue . . .
```

CHAPTER 4 : OPERATION

In our project, there are 10 functions that we have used. The first function is int main() function. The main() function is where the program execution begins. After that, all of the other 10 functions are in the main function(). The program will start with the with the system will ask user to choose what they want to do in the system, whether to input, display, search, sort , delete, or update. As usual, the user will input first their data since there are no data to display when they first enter the program. In this process, there will be menu(Information data) function to use.In this function the user will input their student data since it include student home sport and the sport that they enter. For next choice which is display() function will be used. In this function ,to display the data in the linked list, we need to assign head pointer to the current pointer in order to get the data. Then, the program will just display according to the data that already input.

Next, in the searching process, there are 2 function that used which is search() function and displayS() function. search() function is used to find the position of the data in the linked list. Next displayS() function is used when the data is found, then the system can display the data that the user search. In this process, the user can search their data by entering their student id number so that they can check they student that already store in the data.Furthermore, the user also can sort their data in the sort process. There also have 2 function that need to be used in this process which is sort() function and Sorted() function. In the sort() function, the system will sort the in ascending order. Then in the Sorted() function, the sort() function will be called back and this function will display the data already sorted.Next the process will continue if the user want to delete a data. In the deleted() function, it require the search() function in this process because the system need to find the location of the data first in order to delete it. After the system found the data, the data will be free or delete in the deleted() function.

For the last option which is update, there are 5 option that the user can update, which is Id, Name, Class, HomeSport, SportEvent. In this process the user need to input the searching Id so that the system can locate the data. Then, if the user choose between the 5 option, the system will detect the option chosen then will update the data. Lastly, the system will be end if the user choose to end the program in the first function which is number 7 or labeled 'End Program'.

CONCLUSION

After we have spent the allocated time to finish this data structure project. We gained a lot of knowledge about the C programming language's data structure and were able to know its advantages and disadvantages in terms of simplicity of learning and usefulness.We have learned many functions and they are very helpful in the production of our project. We learned the skills required to build structs that store a number of field members using an array of structures. We also had the chance to determine the number of records and sort the data in ascending order, as well as learn how to add, display, search, edit, and delete records. Additionally, we were able to learn about the memory and storage requirements of functions, how to define initial data variables, and the proper names to use for variables in structural arrays that cannot be preceded by special characters, no spaces, or keywords. With all of the knowledge that we have implemented in this project, this program functions flawlessly.