Student Enrollment System

- [NOTE] Use arrow keys to navigate the program.
- [NOTE] Maximize the window to avoid glitches.
- [NOTE] Student's Age must be 6 to 60 yrs old only.
- [NOTE] Student's ID Number must have 5 length only.
- [NOTE] Student's First Name, Middle Name, and Last Name is only 15 characters long
- [NOTE] Student's Address is only 20 characters long
- [NOTE] Contact Number, 9-11 digit only.

Global Functions

1. bool compareAlpha(Pair &, Pair &) - Compares alphabets, used for vector sorting in printing functions.
2. bool compareNum(Pair &, Pair &) - Compares numbers, used for vector sorting in printing functions.
3. void gotoxy(int,int) - Used for console design.
4. void color(int) - Used for design.

Struct

Struct Name: Pair

Attributes:

1. int first - The student ID Number.
2. Student second - The student class as the second pair.

Classes

Class Name: Student

Functions:

1. void inputInfos(int, int) - Used for data inputing and data editing.
    - first parameter: ID Number/Key
    - second parameter: Type of data to Input; 1 = Last Name, 2 = First Name, and so on. 8 = input all.
2. void printInfos() - Prints the students' informations.
3. string wMyFamName() - Returns the student's family name, used in sorting.
4. string wMyGender() - Returns the student's gender, used in sorting.

Attributes:

1. details[8] - an array of string
   - details[0] - ID Number
   - details[1] - Last Name
   - details[2] - First Name
   - details[3] - Middle Name
   - details[4] - Gender
   - details[5] - Age
   - details[6] - Contact Number
   - details[7] - Address

Class Name: HashTable

Functions:

1. bool isEmpty() - Checks if empty.
2. int hashFunc(int) - Gets the key modulo number of hash groups.
   - first parameter - the key or ID number
3. void enrollStud(int) - Insert student record and create student object.
   - first parameter - the key or ID number
4. void searchStud(int) - Search for student record using ID number.
   - first parameter - the key or ID number
5. void unenrollStud(int) - Removes student record using ID number.
   - first parameter - the key or ID number
6. void printType(int) - Wrapper for printing functions.
   - first parameter - type of printing, printSort or printGender
7. void editInfos(int, int) - Finds and edits student record.
   - first parameter - the key or ID number
   - second parameter - type of data to edit; 1 = Last Name, 2 = First Name and so on, 8 = edit all except the student 's ID
8. void printSort(int) - Prints students' record alphabetically or based on ID number.
   - first parameter - type if sort
9. void printGender(string) - Prints students' record according to gender.
   - first parameter - gender type
10. void fileRead(string) - Used to read a text file for student's records.
    - first parameter - file name

Attributes:

1. static const int hGroups = 13 - The static constant number of hash groups.
2. LinkedList< Pair > table[hGroups] - The list of pair int and student class.

Class Name: Helper

Functions:

1. void clear_screen() - Clears the screen, and prompts to continue.
2. void pause() - Pauses the program until further input.
3. bool inputCheck(int, string) - Checks the student information inputs.
4. void reformatInfo(string&) - Reformats the information of student into camel case.
5. int returnIntNumInput() - Used for inputing and checking; return safe integer number.
6. string returnStringNumInput() - Used for inputing and checking; return string that is number.
7. string returnStringWordInput() - Used for inputing and checking; return string that is a word.
8. void box() - For design
9. int menu(int) - For menu printing and navigation.
   - first paramenter - Fype of menu
10. void welcome() - For instructions and description.
11. int StudentID() - Prompts for the student ID plus checker.
12. void infoHeader() - For table printing.

Class Name: LinkedList

Functions:

1. void insertEnd(T) - Used for student enrollment
2. void insertBeginning(T val) - Makes insert end work better
   - first parameter - The value with templated data type.
3. void deleteBeginning() - Used for unenrolling a student.
4. void deleteEnd() - Used for unenrolling a student.
5. void deleteAtPos(int) - Used for unenrolling a student.
   - first parameter - position
6. bool findNode(int) - Checks if student id is present.
   - first parameter - student ID
7. int findNodePos(int) - Useful for unenrolling a student.
   - first parameter - student ID
8. bool isEmpty() - Checks if linked list is empty.
9. int size() - Useful for checking if hash is empty and how many student is enrolled.
