Test Run:

```
[marti540@empress cs211]$ ./a.out
Enter student info? Y or N :n
Theres no students.
Bye
[marti540@empress cs211]$ ./a.out
Enter student info? Y or N :yy
yy is invalid. Enter Y or N: 100
100 is invalid. Enter Y or N: a
a is invalid. Enter Y or N: y
Enter student id: mike
Invalid. Enter a score between 0 and 100 only: 0
Invalid. Enter a score between 0 and 100 only: 100000
Invalid. Enter a score between 0 and 100 only: 100
Enter student name: mike smith
Enter student gender: mike
mike is invalid. Enter M or F: 100
100 is invalid. Enter M or F: 1
1 is invalid. Enter M or F: a
a is invalid. Enter M or F: m
Enter student major: computer science
Enter student age : mike
Invalid. Enter a score between 0 and 100 only: 151
Invalid. Enter a score between 0 and 100 only: -1
Invalid. Enter a score between 0 and 100 only: 20
Enter student info? Y or N: y
Enter student id: 200
Enter student name: kathy ross
Enter student gender: f
Enter student major: bio tech
Enter student age : 22
Enter student info? Y or N: y
ARRAY IS FULL!!!
Enter the id you are looking for: 200
**************************
Student id: 200
Student name: kathy ross
Student gender: F
Student major: bio tech
student age : 22
**************************
****************************************
The youngest student is 20 years old
The oldest student is 22 years old
****************************************
ID       Name          Gender    Major     Age
100       mike smith     M         computer science20
200       kathy ross     F         bio tech  22
[marti540@empress cs211]$
```

lab1-struct.C

```
#include <iostream>
#include <iomanip>
#include "inputCheck.h"
using namespace std;

struct student
{
  int id;
  string name;
  char gender;
  string major;
  int age;
};


int getStuData(student a[]);
int findID(int num,int look,student a[]);
void displayStu(student a);
void findYoungOld(student a[],int num ,int& young,int& old);
void displayAllStu(student a[],int num);
void displayStud(student a);

const int MAX = 2;

int main()
{
  student ar[MAX];
  int num = getStuData(ar);
  int look;

  if(num == 0)
    {
      cout << "Theres no students." << endl;
      cout << "Bye" <<endl;
      exit(0);
    }
    cout << "Enter the id you are looking for: ";
    cin >> look;


    int found = findID(num,look,ar);
    int young = 999;
    int old = -1;

    findYoungOld(ar,num,young,old);

    cout << "****************************************" << endl;
    cout << "The youngest student is " << young << " years old" << endl;
    cout << "The oldest student is " << old << " years old" << endl;
    cout << "****************************************" << endl;

    displayAllStu(ar,num);

  return 0;
}
```

```
int getStuData(student a[])
{
 int i=0;
 string msg1= "Enter Y or N: ";
 string msg2= "Enter M or F: ";
 cout << "Enter student info? Y or N :";
 char ans = getResponse(msg1);

      while (ans == 'y' || ans == 'Y')
        {
         if(i>=MAX)
           {
            cout << "ARRAY IS FULL!!!" << endl;
            ans= ' ';
           }
         else
           {
            cout << "Enter student id: ";
           a[i].id = getNumberInRange(1, 9999, "Invalid. Enter a score between 0 and 100 only: ");
            cin.ignore();
            cout << "Enter student name: ";
            getline(cin,a[i].name);
            cout << "Enter student gender: ";
            a[i].gender = getGender(msg2);
            cout << "Enter student major: ";
            cin.ignore();
            getline(cin,a[i].major);
            cout << "Enter student age : ";
            a[i].age = getNumberInRange(0, 150, "Invalid. Enter a score between 0 and 100 only: ");
            cout << "Enter student info? Y or N: ";
            ans= getResponse(msg1);
            i++;
           }
        }

 student Default = {0, " ", ' ', " ", 999}; //defalut values

 for(int x = i; x < MAX; x++)
   {
    a[x] = Default;
   }


return i;
}

int  findID(int num,int look,student a[])
{
 for(int i=0; i< num;i++)
   {
    if(look == a[i].id)
        {
         displayStu(a[i]);
         return i;
        }
   }
 cout << "Id was not found" << endl;
```

```
}

void displayStu(student a)
{
  cout << "****************************" << endl;
  cout << "Student id: " << a.id << endl;
  cout << "Student name: " << a.name << endl;
  cout << "Student gender: " << a.gender << endl;
  cout << "Student major: " << a.major << endl;
  cout << "student age : " << a.age << endl;
  cout << "****************************" << endl;

}

/*This function will return the youngest and oldest ages */
void  findYoungOld(student a[],int num,int& young,int& old)
{
  for(int i=0; i < num;i++)
    {
      if(a[i].age < young)
           {
            young = a[i].age;
           }
      if(a[i].age > old)
           {
            old = a[i].age;
           }
    }
}
//DISPLAY ALL STUDENTS
void displayAllStu(student a[],int num)
{
  cout << setw(10)<< left << "ID" << setw(15) << "Name" << setw(10) << "Gender" << setw(10) << "Major" <<
setw(10) << "Age" << endl;
  for(int i= 0;i < num; i++)
    {
      displayStud(a[i]);
    }
}
void displayStud(student a)
{
  cout <<  setw(10) << a.id << setw(15) << a.name << setw(10) << a.gender << setw(10) << a.major << setw(10)<<
a.age << endl;
}
```

inputCheck.h

```
#include <iostream>
#include <string>
#include <cstring>
#include <stdio.h>
#include <stdlib.h>
using namespace std;
//prototypes here
char getLetter(string msg);
char getResponse(string msg);
char getGender(string msg);
int getNumberInRange(int low, int high, string msg);

//CHECKS RESPONSE FOR GENDER & Y OR N
char getLetter(string msg)
{
 string input;
 cin >> input;

 while(input.length() != 1 || !isalpha(input[0]))
   {
    cout << input << " is invalid. " << msg;
    cin >> input;
   }
 return input[0];
}
char getResponse(string msg)//y or n
{
 char  ch;
 ch = getLetter(msg);

 while(toupper(ch) != 'Y' && toupper(ch) != 'N')
   {
    cout << ch <<  " is invalid. " <<  msg;
    ch = getLetter(msg);
   }
 return toupper(ch);
}
char getGender(string msg)//m or f
{
 char  ch;
 ch = getLetter(msg);

 while(toupper(ch) != 'F' && toupper(ch) != 'M')
   {
    cout << ch <<  " is invalid. " <<  msg;
    ch = getLetter(msg);
   }
 return toupper(ch);
}
int getNumberInRange(int low, int high, string msg)
{
 string input;
 bool nonNumber;
 bool invalid;
 int inpt;
 cin >> input;
 do
```

```
  {

     char * cstr =new char[input.length()+1];
    strcpy (cstr, input.c_str());
    inpt = atoi(cstr);
    nonNumber = false;
    invalid = false;
    //check each character in "input" and if a non-digit character is found, set "nonNumber" to true
    for(int i = 0; i < input.length(); i++)
      {
       if(!isdigit(cstr[i]))
         {
          nonNumber = true;
         }

      }



    if(nonNumber == true || inpt > high || inpt < low)
      {

          invalid = true;
      }

    //if .invalid. is true, ask for a new input
    if(invalid) //same as invalid == true
      {
          cout << msg;
          cin >> input;

      }

 }while(invalid);
 //returning a valid integer between low and high
 return inpt;
}
```