

STRUCTURED PROGRAMMING

Username: ebazan089

SECTION I: Algorithm/code in C++

QUESTION 1: DESIGNING THE ALGORITHM /FLOWCHART

Create the algorithm (pseudo-code) that will ask the user to choose what kind of conversion he/she likes to do (pounds to kilograms or kilograms to pounds).

There are 2.2046 pounds in a kilogram, 1000 grams in a kilogram, and 16 ounces in a pound.

If the first option was selected, enter a weight in pounds and ounces and outputs the equivalent weight in kilograms and grams or to enter 0 if wants to find out the max, the min and the average of the weights the user enter as far.

If the second option was selected, enter a weight in kilograms and grams and outputs the equivalent weight in pounds and ounces or to enter 0 if wants to find out the max, the min and the average of the weights the user enter as far.

Include a loop that lets the user repeat this computation for new input values until the user says he or she wants to end the program (the 3th option of the main menu).

1. START

2. Ask the user ENTER one of the three options.
3. If the user choose 1, it will convert pounds to kilograms. **Go STEP 6**
4. If the user choose 2, it will conver kilograms to pounds. **Go STEP 7**
5. If the user choose 3, it will leave the program. **Go STEP 9**
6. User enter the value WEIGHT in POUNDS, then the program will convert this value to KILOGRAMS and it will save in the table. The user can enter until 15 values or it stops when the user enter 0. **Go STEP 8**
7. User enter the value WEIGHT in KILOGRAMS, then the program will convert this value to POUNDS and it will save in the table. The user can enter until 15 values or it stops when the user enter 0. **Go STEP 8**
8. The program will show all the values that the user enter, also it will show the MAXIMUM, MINIMUM and AVERAGE. **Go STEP 2**

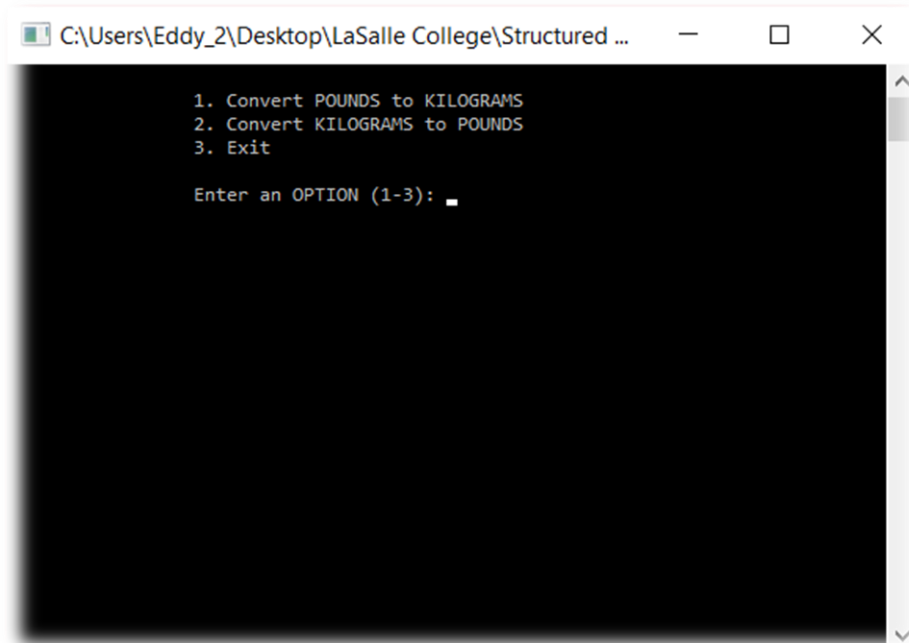
9. END

QUESTION 2: TRANSLATING ALGORITHM INTO A C++

VARIABLES	DESCRIPTION
<code>int ans, i, x, opt;</code>	Global variables type INTEGER
<code>double value, maxV=0, minV=0, averageV=0;</code>	Global variables type DOUBLE
<code>const int SIZE = 15;</code>	Constant INTEGER with value 15
<code>double table[SIZE], values[SIZE];</code>	Array type DOUBLE with the size of the constant SIZE

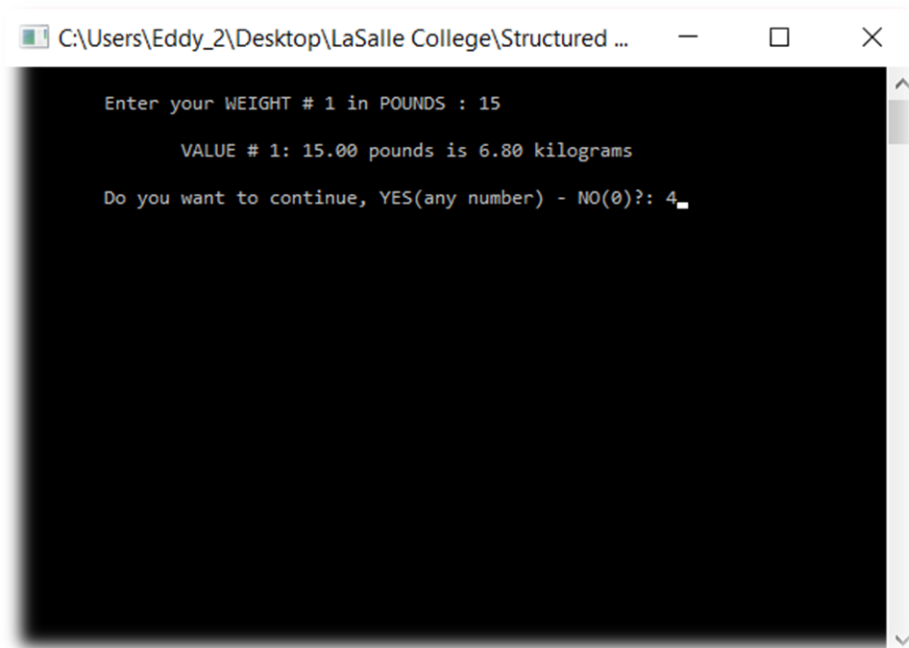
FUNCTION NAME	DESCRIPTION
<code>void pounds_kilograms(double table[]);</code>	Calculate the value from pounds to kilograms and save it in the table.
<code>void kilograms_pounds(double table[]);</code>	Calculate the value from kilograms to pounds and save it in the table.
<code>void maxValue(double table[], int tsize, int opt);</code>	Calculate the MAXIMUM value with all the values of the table.
<code>void minValue(double table[], int tsize, int opt);</code>	Calculate the MINIMUM value with all the values of the table.
<code>void averageValue(double table[], int tsize, int opt);</code>	Calculate AVERAGE value with all the values of the table.

MENU



```
C:\Users\Eddy_2\Desktop\LaSalle College\Structured ...  
1. Convert POUNDS to KILOGRAMS  
2. Convert KILOGRAMS to POUNDS  
3. Exit  
  
Enter an OPTION (1-3):
```

OPTION 1: CONVERT POUNDS TO KILOGRAMS



```
C:\Users\Eddy_2\Desktop\LaSalle College\Structured ...  
  
Enter your WEIGHT # 1 in POUNDS : 15  
  
VALUE # 1: 15.00 pounds is 6.80 kilograms  
  
Do you want to continue, YES(any number) - NO(0)?: 4
```

**SHOW MAXIMUM, MINIMUM AND AVERAGE VALUE, ALSO ALL THE
VALUES CONVERTED KILOGRAMS**

```
C:\Users\Eddy_2\Desktop\LaSalle College\Structured ...

Enter your WEIGHT # 15 in POUNDS : 11.2

VALUE # 15: 11.20 pounds is 5.08 kilograms

The MAXIMUM weight is 40.64 kilograms

The MINIMUM weight is 0.81 kilograms

The AVERAGE weight is 19.78 kilograms

TABLE SIZE is 15
Value #1 = 12.00 pounds = 5.44 kilograms
Value #2 = 23.50 pounds = 10.66 kilograms
Value #3 = 34.60 pounds = 15.69 kilograms
Value #4 = 45.30 pounds = 20.55 kilograms
Value #5 = 1.78 pounds = 0.81 kilograms
Value #6 = 23.70 pounds = 10.75 kilograms
Value #7 = 45.30 pounds = 20.55 kilograms
Value #8 = 87.60 pounds = 39.74 kilograms
Value #9 = 69.30 pounds = 31.43 kilograms
Value #10 = 89.60 pounds = 40.64 kilograms
Value #11 = 45.10 pounds = 20.46 kilograms
Value #12 = 34.70 pounds = 15.74 kilograms
Value #13 = 76.46 pounds = 34.68 kilograms
Value #14 = 53.80 pounds = 24.40 kilograms
Value #15 = 11.20 pounds = 5.08 kilograms
Press any key to continue . . .
```

OPTION 2: CONVERT KILOGRAMS TO POUNDS

```
C:\Users\Eddy_2\Desktop\LaSalle College\Structured ...

Enter your WEIGHT # 1 in KILOGRAMS = 5.4

VALUE # 1: 5.40 kilograms is 11.90 pounds

Do you want to continue, YES(any number) - NO(0)? : 3
```

**SHOW MAXIMUM, MINIMUM AND AVERAGE VALUE, ALSO ALL THE
VALUES CONVERTED POUNDS**

```
C:\Users\Eddy_2\Desktop\LaSalle College\Structured ...

Enter your WEIGHT # 15 in KILOGRAMS = 0.56

VALUE # 15: 0.56 kilograms is 1.23 pounds

The MAXIMUM weight is 198.39 pounds

The MINIMUM weight is 1.23 pounds

The AVERAGE weight is 198.42 pounds

TABLE SIZE is 15
Value #1 = 12.30 kilograms = 27.12 pounds
Value #2 = 12.60 kilograms = 27.78 pounds
Value #3 = 11.00 kilograms = 24.25 pounds
Value #4 = 23.70 kilograms = 52.25 pounds
Value #5 = 33.70 kilograms = 74.30 pounds
Value #6 = 10.50 kilograms = 23.15 pounds
Value #7 = 9.87 kilograms = 21.76 pounds
Value #8 = 34.60 kilograms = 76.28 pounds
Value #9 = 33.80 kilograms = 74.52 pounds
Value #10 = 54.60 kilograms = 120.37 pounds
Value #11 = 67.20 kilograms = 148.15 pounds
Value #12 = 77.00 kilograms = 169.75 pounds
Value #13 = 89.99 kilograms = 198.39 pounds
Value #14 = 67.50 kilograms = 148.81 pounds
Value #15 = 0.56 kilograms = 1.23 pounds
Press any key to continue . . .
```

OPTION 3: EXIT

```
C:\Users\Eddy_2\Desktop\LaSalle College\Structured ...

1. Convert POUNDS to KILOGRAMS
2. Convert KILOGRAMS to POUNDS
3. Exit

Enter an OPTION (1-3): 3

Now you are leaving the APPLICATION...!!!
Press any key to continue . . .
```

QUESTION 3-1: CREATE A STRUCTURE

Given the following data structure, declare a structure named *Courses* with the appropriate fields.

Course Number	Title	Hours per week	Session
420-P16-AS	Structured Programming	6	Fall 2016

```
struct Session{
    string season;
    int year;
};

struct Courses{
    string courseNumber;
    string titleC;
    int hoursC;
    Session termC;
};
```

QUESTION 3-2: RUN AND DISPLAY THE OUTPUT

Write a code program in C++ that allows entering all the courses data that you have for current session at compile-time using the structure from previous point. Present the output you obtain in the documentation of the project (print-screens) or by copy-paste as comment in C++ file.

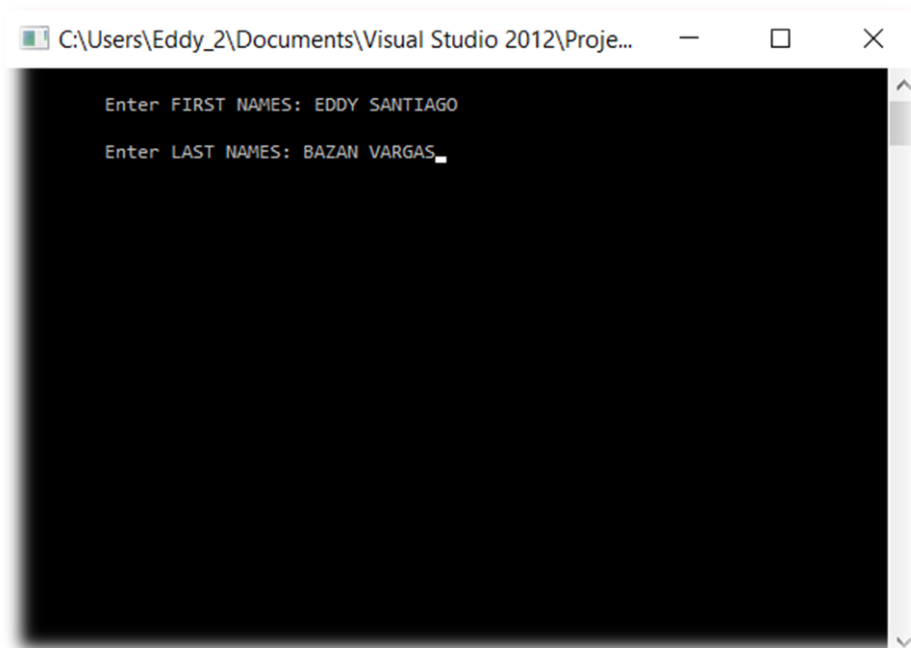
STRUCTURE	DESCRIPTION
<pre>struct Phone{ int code; int area; int part1; int part2; };</pre>	Structure type called <i>Phone</i> with parameters integer <i>code</i> , <i>area</i> , <i>part1</i> , <i>part2</i>
<pre>struct Person{ int numberCode; string firstName; string lastName; string email; Phone phoneNumber; };</pre>	Structure type called <i>Person</i> with parameters integer <i>numberCode</i> , string <i>firstName</i> , <i>lastName</i> , <i>email</i> and structure <i>Phone phoneNumber</i>

VARIABLES	DESCRIPTION
<pre>int i, codeNumber, optC;</pre>	Global variables type INTEGER
<pre>vector <Person> table(0);</pre>	Vector type PERSON (Structure)

<code>Person data;</code>	Variable type PERSON (Structure)
<code>Phone test;</code>	Variable type Phone (Structure)

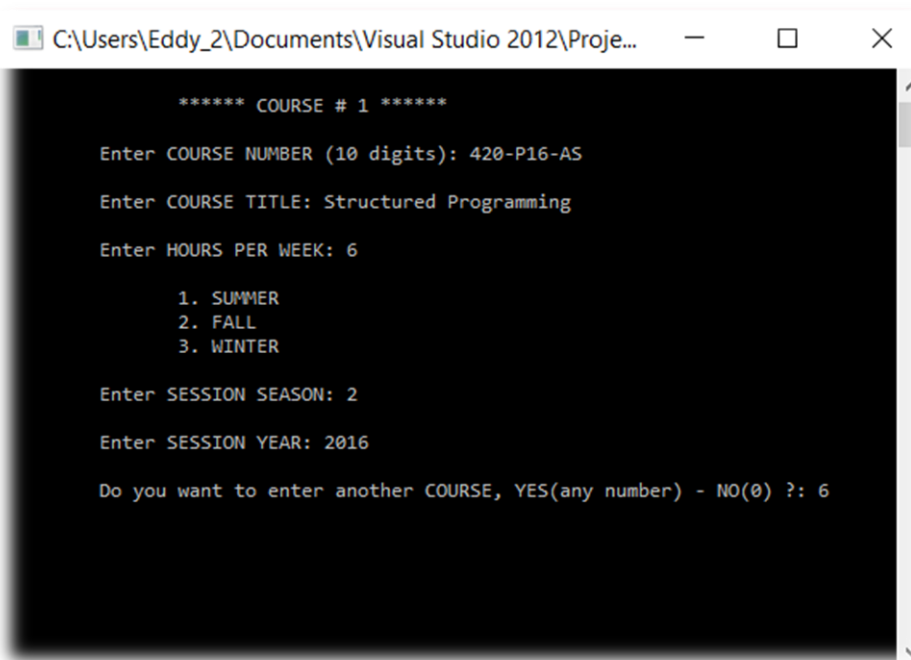
FUNCTION NAME	DESCRIPTION
<code>void CreateAddStudent(vector <Person>& table, Person data);</code>	Create and add a new student to the vector
<code>void SearchStudent(vector <Person>& table);</code>	Search a student with his CODE NUMBER in the vector
<code>void DisplayStudent(vector <Person>& table);</code>	Show all the students in the vector
<code>int validateCodeNumber(vector <Person>& table, int value);</code>	Return 0 if the code number doesn't exist in the vector. Return 1 if the code number exists in the vector.
<code>Phone insertPhone();</code>	Return a value type Phone that the user insert the phone number
<code>Phone AreaAndNumber();</code>	Return a value type Phone with the code area and the the last 7 digits of the number that the user enter

ENTER FIRST AND LAST NAMES FOR A STUDENT



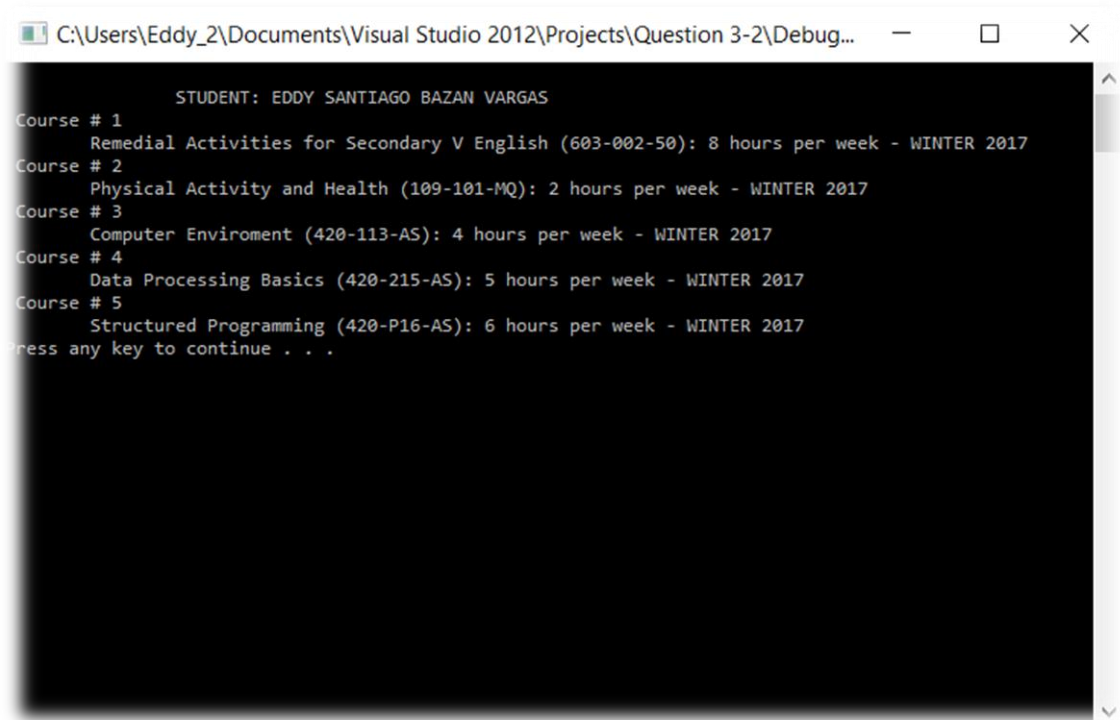
```
C:\Users\Eddy_2\Documents\Visual Studio 2012\Proje...  
Enter FIRST NAMES: EDDY SANTIAGO  
Enter LAST NAMES: BAZAN VARGAS_
```

ENTER CORUSE INFORMATION



```
C:\Users\Eddy_2\Documents\Visual Studio 2012\Proje...  
***** COURSE # 1 *****  
Enter COURSE NUMBER (10 digits): 420-P16-AS  
Enter COURSE TITLE: Structured Programming  
Enter HOURS PER WEEK: 6  
1. SUMMER  
2. FALL  
3. WINTER  
Enter SESSION SEASON: 2  
Enter SESSION YEAR: 2016  
Do you want to enter another COURSE, YES(any number) - NO(0)?: 6
```


SHOW ALL COURSES THAT THE STUDENT HAVE IN THE SEMESTER



```
STUDENT: EDDY SANTIAGO BAZAN VARGAS
Course # 1
Remedial Activities for Secondary V English (603-002-50): 8 hours per week - WINTER 2017
Course # 2
Physical Activity and Health (109-101-MQ): 2 hours per week - WINTER 2017
Course # 3
Computer Enviroment (420-113-AS): 4 hours per week - WINTER 2017
Course # 4
Data Processing Basics (420-215-AS): 5 hours per week - WINTER 2017
Course # 5
Structured Programming (420-P16-AS): 6 hours per week - WINTER 2017
Press any key to continue . . .
```

SECTION II: Programming in C++

Create an application that keeps track of students, knowing that a student is defined by: a number, a first name, a last name, an email and a phone number where first name and last name are defined as a structure *Person* and phone number is a structure *Phone* with the following properties:

- 1- international code : 1digit
- 2- country code : 2 digits
- 3- regional code : 3 digit
- 4- home code : 7 digits

The menu is as follows:

Student Management Application

- 1. Create and add a student.
- 2. Search for a student by number
- 3. Display the student list
- 4 Exit the application

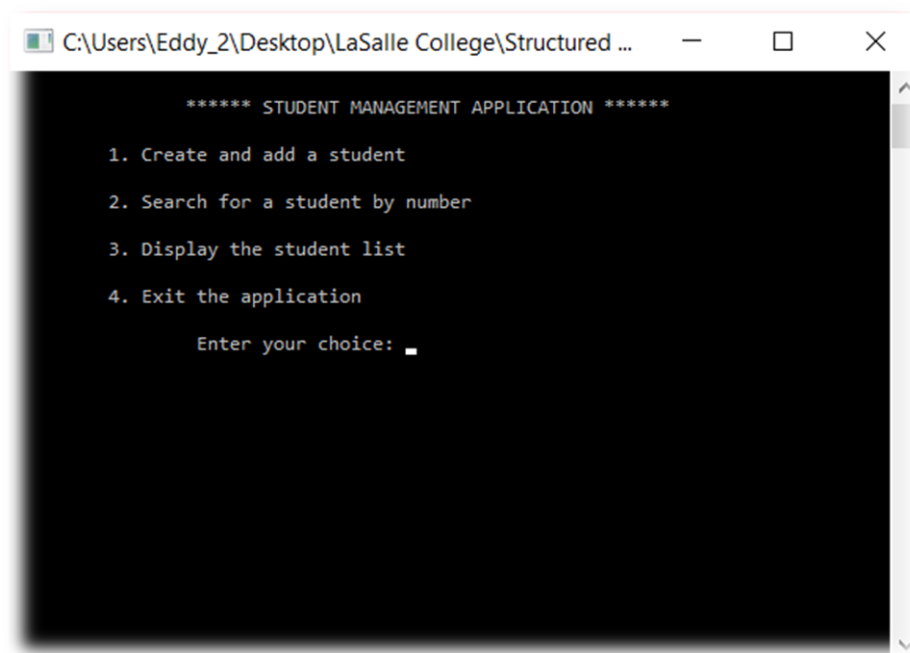
Enter your choice: _

VARIABLES	DESCRIPTION
<code>int i, opt;</code>	Global variables type INTEGER

<code>const int SIZE=10;</code>	Constant INTEGER with value 15
<code>string FirstNames, LastNames, seasonS;</code>	Global variables type STRING
<code>vector <Courses> tableS(0);</code>	Vector type Courses (Structure)

FUNCTION NAME	DESCRIPTION
<code>string insertCourseNumber();</code>	Return a value <i>String</i> with the value of de Course Number that the user enter
<code>Session term();</code>	Return a value <i>Session</i> (Structure) with the value of de term that the user enter
<code>void displayCourses(vector <Courses>& table);</code>	Show all the courses that the user enter

QUESTION 4: DISPLAY THE STUDENT MANAGEMENT MENU



```

C:\Users\Eddy_2\Desktop\LaSalle College\Structured ...
***** STUDENT MANAGEMENT APPLICATION *****

1. Create and add a student
2. Search for a student by number
3. Display the student list
4. Exit the application

Enter your choice: _

```

QUESTION 5: INPUT STUDENT INFORMATION USING STRUCTURE

```
C:\Users\Eddy_2\Desktop\LaSalle College\Structured ...

***** CREATE AND ADD A NEW STUDENT *****

Insert STUDENT CODE (7 digits): 1710793

Insert the FIRST NAME: EDDY

Insert the LAST NAME: BAZAN

Insert an EMAIL: ebazan@hotmail.com

Insert a PHONE NUMBER:

    1. International Code
    2. Country Code
    3. Regional Code
    4. Home Code

Enter a choice: 3

Insert your AREA CODE (3 digits): 514

Insert your NUMBER PART 1 (3 digits): 577

Insert your NUMBER PART 2 (4 digits): 3728

***** STUDENT # 1 *****
CODE:1710793
NAMES: EDDY BAZAN
EMAIL: ebazan@hotmail.com
PHONE NUMBER: (514) - 577 - 3728
Press any key to continue . . .
```

QUESTION 6: SEARCH FOR STUDENT BY STUDENT NUMBER

```
C:\Users\Eddy_2\Desktop\LaSalle College\Structured ...

***** SEARCH A STUDENT BY CODE NUMBER *****
Insert STUDENT CODE to SEARCH: 1715679

***** STUDENT # 3 *****
CODE:1715679
NAMES: MARCOS CAICEDO
EMAIL: mcaicedo@yahoo.com
PHONE NUMBER: (514) - 234 - 3241

Do you want to continue YES(any number) or NO(1):
```

QUESTION 7: DISPLAY STUDENT LIST

```
C:\Users\Eddy_2\Desktop\LaSalle College\Structured ...  -  □  X

***** STUDENT MANAGEMENT APPLICATION *****

1. Create and add a student
2. Search for a student by number
3. Display the student list
4. Exit the application

Enter your choice: 3

***** STUDENT # 1 *****
CODE:1710793
NAMES: EDDY BAZAN
EMAIL: ebazan@hotmail.com
PHONE NUMBER: (514) - 577 - 3728

***** STUDENT # 2 *****
CODE:1720543
NAMES: CARLOS MORENO
EMAIL: cmoreno@gmail.com
PHONE NUMBER: +2(438) - 234 - 1234

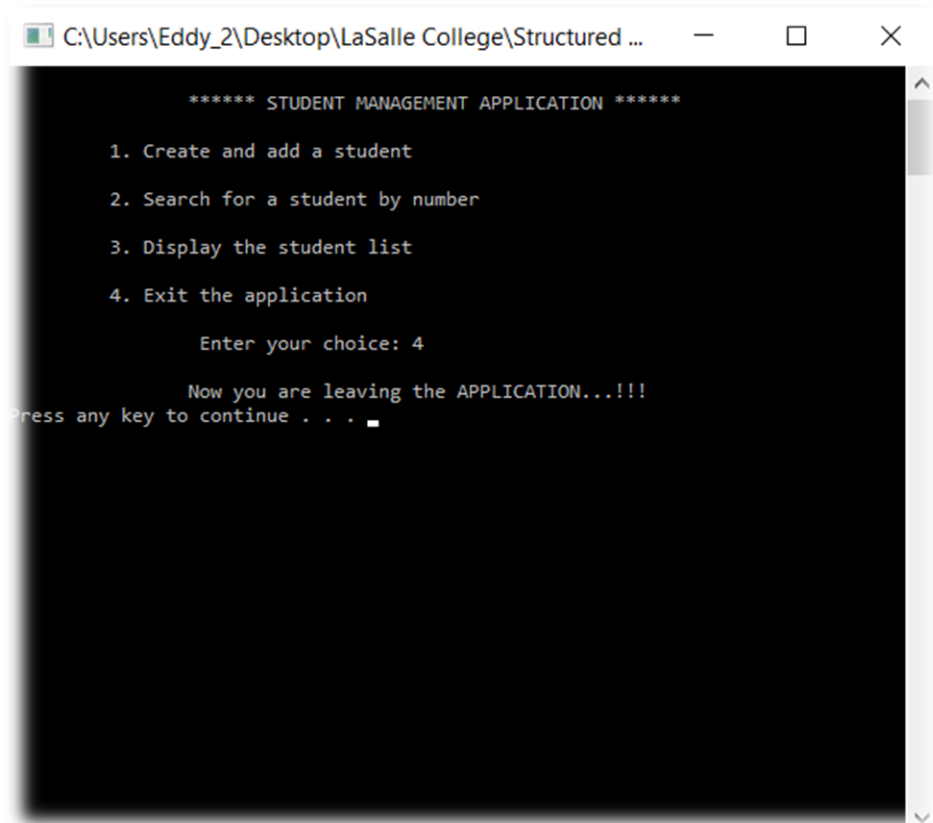
***** STUDENT # 3 *****
CODE:1715679
NAMES: MARCOS CAICEDO
EMAIL: mcaicedo@yahoo.com
PHONE NUMBER: (514) - 234 - 3241

***** STUDENT # 4 *****
CODE:1710100
NAMES: EDUARDO DIAZ
EMAIL: ediaz@gmail.com
PHONE NUMBER: +23(438) - 234 - 1256

***** STUDENT # 5 *****
CODE:1720235
NAMES: MELISSA SARCOS
EMAIL: msarcos@yahoo.com
PHONE NUMBER: +98(438) - 234 - 8769

Press any key to continue . . .
```

QUESTION 8: EXIT THE APPLICATION



A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\Eddy_2\Desktop\LaSalle College\Structured ... The window contains the following text:

```
***** STUDENT MANAGEMENT APPLICATION *****  
  
1. Create and add a student  
2. Search for a student by number  
3. Display the student list  
4. Exit the application  
  
Enter your choice: 4  
  
Now you are leaving the APPLICATION...!!!  
Press any key to continue . . .
```