Vector Graphic

**Due Time:** 23.59, 25 September 2016 **Earnings:** 4% of your final grade

***NOTE: The code in this assignment must be your own work. It must not be code taken from another student or written for you by someone else, even if you give a reference to the person you got it from (attribution); if it is not entirely your own work it will be treated as plagiarism and given a fail mark, or less.***

**Purpose:** You are to write the code in Visual Studio 2013 for a simple C language console application that holds the data of a Vector Graphics application (there is no actual graphics in the assignment) using dynamic memory allocation for its data. This will give you an opportunity to review material that has already been taught in earlier courses and get up to speed in C programming that will be needed for C++. Part of the code (minus the usual headers) is shown on the next page. You **MUST** use this code **without modification (not a single character changed): no code added or removed, no macros, no defines and no statics**. Your task is to implement only the functions that are declared and not add any new ones. All your code is in a single file named ass0.cpp.

The Vector Graphic is an array of Graphic Elements, each of which is made of a number of Lines between Points.

In this assignment, when the application is running the user can

* Add a new Graphic Element to the Vector Graphic
* Print all the details of the Graphic Elements in the Vector Graphic

**An example of the output of the running application is given at the end. Yours must work identically and produce identical output.**

Note the following:

* The file must be named ass0.cpp.
* You cannot use any C++ constructs or classes. You must only use malloc() and free() (not realloc) for dynamic memory management
* You can only use functions like strlen() and strcpy() or similar etc. from the standard C library to handle strings (you cannot use the C++ string class).
* When the application terminates it releases **all** dynamically allocated memory so it does not have a resource leak (or you lose 30%).

See the Marking Sheet for how you can lose marks, but you will lose 60% if:

1. you change the supplied code in any way at all (not a single character) - no code added (except header includes) or removed, no macros, no defines, no statics and no additional functions,

2. it fails to build in Visual Studio 2013,

3. It crashes in normal operation (such as printing from an empty list etc.),

4. it doesn’t produce the example output.

Part of the code is shown on the next page. You MUST use this code **without modification.** Your task is to add the implementation of the functions that are declared using the style of the posted Submission Standard. All the code is in a single file named ass0.cpp.

**What to Submit :** Use Blackboard to submit this assignment as a zip file (**not** RAR) containing only the single source code file (ass0.c). If you are taking both CST8219 and CST8233, you need only to submit to CST8219. The name of the zipped folder **must** contain your name as a prefix so that I can identify it, for example, for CST8219, using my name the file would be tyleraAss0CST8219.zip. It is also vital that you include the Cover Information (as specified in the Submission Standard) as a file header in your source file so the file can be identified as yours. Use comment lines in the file to include the header. **Before you submit the code, check that it builds and executes in Visual Studio 2013 as you expect - if it doesn’t build for me, for whatever reason, you get a deduction of at least 60%**. There is a late penalty of 25% per day. Don’t send me the file as an email attachment – it will get 0.

***Example code – I will use it to mark your assignment. Don’t change or add to it (not even a single character), but include the usual headers. This code is also in a text file on BlackBoard, so you don’t need to type it:***

enum{ RUNNING = 1 };

struct Point

{

int x, y;

};

struct Line

{

Point start;

Point end;

};

struct GraphicElement

{

enum{ SIZE = 256 };

unsigned int numLines;

Line\* pLines;

char name[SIZE];

};

typedef struct

{

unsigned int numGraphicElements;

GraphicElement\* pElements;

}VectorGraphic;

void InitVectorGraphic(VectorGraphic\*);

void AddGraphicElement(VectorGraphic\*);

void ReportVectorGraphic(VectorGraphic\*);

void CleanUpVectorGraphic(VectorGraphic\*);

VectorGraphic Image;

int main()

{

char response;

InitVectorGraphic(&Image);

while (RUNNING)

{

printf("\nPlease select an option:\n");

printf("1. Add a Graphic Element\n");

printf("2. List the Graphic Elements\n");

printf("q. Quit\n");

printf("CHOICE: ");

fflush(stdin);

scanf("%c", &response);

switch (response)

{

case '1':AddGraphicElement(&Image); break;

case '2':ReportVectorGraphic(&Image); break;

case 'q':CleanUpVectorGraphic(&Image); return 0;

default:printf("Please enter a valid option\n");

}

printf("\n");

}

}

Example Output:

Please select an option:

1. Add a Graphic Element

2. List the Graphic Elements

q. Quit

CHOICE: 1

ADDING A Graphic Element

Please enter the name of the new GraphicElement(<256 characters): triangle

How many lines are there in the new GraphicElement? 3

Please enter the x coord of the start point of line index 0: 0

Please enter the y coord of the start point of line index 0: 0

Please enter the x coord of the end point of line index 0: 1

Please enter the y coord of the end point of line index 0: 1

Please enter the x coord of the start point of line index 1: 1

Please enter the y coord of the start point of line index 1: 1

Please enter the x coord of the end point of line index 1: -1

Please enter the y coord of the end point of line index 1: 1

Please enter the x coord of the start point of line index 2: -1

Please enter the y coord of the start point of line index 2: 1

Please enter the x coord of the end point of line index 2: 0

Please enter the y coord of the end point of line index 2: 0

Please select an option:

1. Add a Graphic Element

2. List the Graphic Elements

q. Quit

CHOICE: 1

ADDING A Graphic Element

Please enter the name of the new GraphicElement(<256 characters): square

How many lines are there in the new GraphicElement? 4

Please enter the x coord of the start point of line index 0: 0

Please enter the y coord of the start point of line index 0: 0

Please enter the x coord of the end point of line index 0: 0

Please enter the y coord of the end point of line index 0: 1

Please enter the x coord of the start point of line index 1: 0

Please enter the y coord of the start point of line index 1: 1

Please enter the x coord of the end point of line index 1: 1

Please enter the y coord of the end point of line index 1: 1

Please enter the x coord of the start point of line index 2: 1

Please enter the y coord of the start point of line index 2: 1

Please enter the x coord of the end point of line index 2: 1

Please enter the y coord of the end point of line index 2: 0

Please enter the x coord of the start point of line index 3: 1

Please enter the y coord of the start point of line index 3: 0

Please enter the x coord of the end point of line index 3: 0

Please enter the y coord of the end point of line index 3: 0

Please select an option:

1. Add a Graphic Element

2. List the Graphic Elements

q. Quit

CHOICE: 2

VectorGraphic Report

Reporting Graphic Element #0

Graphic Element name: triangle

Line #0 start x: 0

Line #0 start y: 0

Line #0 end x: 1

Line #0 end y: 1

Line #1 start x: 1

Line #1 start y: 1

Line #1 end x: -1

Line #1 end y: 1

Line #2 start x: -1

Line #2 start y: 1

Line #2 end x: 0

Line #2 end y: 0

Reporting Graphic Element #1

Graphic Element name: square

Line #0 start x: 0

Line #0 start y: 0

Line #0 end x: 0

Line #0 end y: 1

Line #1 start x: 0

Line #1 start y: 1

Line #1 end x: 1

Line #1 end y: 1

Line #2 start x: 1

Line #2 start y: 1

Line #2 end x: 1

Line #2 end y: 0

Line #3 start x: 1

Line #3 start y: 0

Line #3 end x: 0

Line #3 end y: 0

Please select an option:

1. Add a Graphic Element

2. List the Graphic Elements

q. Quit

CHOICE: