Animation Project in C

Due Time: 23.59, Tuesday Jan 28th, 2020 **Earnings:** 2% of your final grade

NOTE: Plan to finish a few days early to avoid last minute hardware/software holdups for which no allowance is given.

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 2019 for a simple C language console application that holds the data of an animation application (there is no actual animation graphics in the assignment) using a linked list in dynamic memory. This will give you an opportunity to review material that has already been taught in an earlier C course and get up to speed for programming that is used in this course. In the lab you will be shown how to set up a Visual Studio 2019 project and what the application looks like when it's running.

The start of the code is shown on the next page; it is also on Brightspace in a text file that you can copy and paste. Because I will use this code to mark the assignment you must not modify it (not a single character changed): no code added or removed, no new global variables, no new functions, no macros, no defines and no statics. Your task is to implement (after the main() function), using C, only the functions that are declared at the top of the ass0.c file and not add any new ones. All your code is in the file ass0.c.

The Animation is a series of Frame objects held in a single linked list (forward list) in dynamic (heap) memory. This memory model will be adapted to a C++ template in a later assignment. When the application is running you can:

add a new Frame to start of the Animation Frame list
delete the last Frame in the list
edit a selected Frame to change the frame name
report the Animation to show the list of Frame details quit.

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 you submit must be named ass0.c,
dynamic memory management is done with malloc() and free(),
there is never any excess dynamic memory allocated - only allocate exactly what is needed for each string entered by the user
you can only use functions like strlen() and strcpy() etc. from the standard C library to handle strings of null terminated char,
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 at least 60% if:

1. you change the supplied code in any way at all - no code added (except header includes) or removed, no macros, no defines, no statics and no additional global functions or variables,
2. it fails to build in Visual Studio 2019
3. It crashes in normal operation (such as reporting an empty list etc.),
4. it doesn't produce the example output.

What to Submit : Use Brightspace to submit this assignment as a zip file (**not** RAR, not 9zip, not 7 zip) containing only the single source code file (ass0.c). 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 **bahrisAss0CST8219.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 2019 as you expect - if it doesn't build for me, for whatever reason, you get a deduction of at least 60%.

make sure you have submitted the correct file – if I cannot build it because the file is wrong or missing from the zip, even if it's an honest mistake, you get 0 – no compromises.

There is a late penalty of 25% per day. Don't send me the file as an email attachment – it will get 0.

Code you must use (also in a text file on Brightspace you can copy and paste). Don't change it.

```
// ass0.c
#define _CRT_SECURE_NO_WARNINGS
#define _CRTDBG_MAP_ALLOC     // need this to get the line identification
//_CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF|_CRTDBG_LEAK_CHECK_DF); // in main, after local declarations
//NB must be in debug build

#include <crtdbg.h>
#include <stdio.h>
#include <string.h>

typedef enum{FALSE = 0, TRUE} BOOL;

struct Frame{
          char* frameName;
          struct Frame* pNext;
};

typedef struct{
          char* animationName;
          struct Frame* frames;
}Animation;

// Forward declarations
void InitAnimation(Animation*);
void InsertFrame(Animation*);
void DeleteFrame(Animation*);
void EditFrame(Animation*);
void ReportAnimation(Animation*);
void CleanUp(Animation*);

int main(void)
{
          char response;
          BOOL RUNNING = TRUE;
          Animation RG;
          _CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF);
          InitAnimation(&RG);

          while (RUNNING)
          {
                    printf( "MENU\n 1. Insert a Frame at the front\n 2. Delete last Frame\n 3. Edit a Frame\n 4. Report the Animation\n 5. Quit\n");
                    scanf("%c", &response);
                    switch (response)
                    {
                    case '1':InsertFrame(&RG);break;
                    case '2':DeleteFrame(&RG);break;
                    case '3':EditFrame(&RG);break;
                    case '4':ReportAnimation(&RG);break;
                    case '5':RUNNING = FALSE;CleanUp(&RG);break;
                    default:printf("Please enter a valid option\n");
                    }
          }
          return 0;
}
```

Example Output:

```
Please enter the Animation name:Animation_1
MENU
 1. Insert a Frame at the front
 2. Delete last Frame
 3. Edit a Frame
 4. Report the Animation
 5. Quit
1
Insert a Frame in the Animation
Please enter the Frame frameName: Frame_1
MENU
```

CST 8219 – F19 - Assignment #0

1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit

1

Insert a Frame in the Animation

Please enter the Frame frameName: Frame_2

MENU

1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit

4

Animation name is Animation_1

Report the Animation

Image #0, file name = Frame_2

Image #1, file name = Frame_1

MENU

1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit

3

Edit a Frame in the Animation

There are 2 Frame(s) in the list. Please specify the index (<= 1) to edit at : 0 The name of this Frame is Frame_2. What do you wish to replace it with? Frame_3

MENU

1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit

4

Animation name is Animation_1

Report the Animation

Image #0, file name = Frame_3

Image #1, file name = Frame_1

MENU

1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit

2

MENU

1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit

3

Edit a Frame in the Animation

There are 1 Frame(s) in the list. Please specify the index (<= 0) to edit at : 0 The name of this Frame is Frame_3. What do you wish to replace it with? Frame_1

MENU

1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit

4

Animation name is Animation_1

Report the Animation

Image #0, file name = Frame_1

MENU

1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit