Homework # 6

01286131 Object-Oriented Programming

Software Engineering Program,

Department of Computer Engineering,

School of Engineering, KMITL

By

65011277 Chanasorn Howattanakulphong

Object-Oriented Programming Homework #6 **Feb 24th, 2023**

Managing Memory

1. Write functions and programs that modifies values through pointers, verify correctness of all functions with test programs and ensure that all programs run as expected.

1.1) Write a function, `void inverse_numbers(double* v, size_t n)`, that replaces `n` numbers starting from `v` by their negative. **Do not use** any standard library functions.

1.2) Write a function, `void replace(char* s, char c1, char c2)`, that replaces all characters matching `c1` found in the C-style string `s` with a character `c2`. **Do not use** any standard library functions. A C-style string is a zero-terminated array of characters, so if you find a char with the value 0 you are at the end (**stop reading** a char at that point).

For example, replace `'l'` with `'X'` in `"Hello, World!"` will change the string to become `"HeXXo, worXd!"`.

1.3) Write a function, `char* encode_hex(const char* s)`, that copies a C-style string into memory it allocates on the free store from the original by encoding each character using two hex digits. **Do not use** any standard library functions.

For example, encoding `"Hello, world!"` will create a C-style string `"48656C6C6F2C20776F726C6421"`.

```
when n = 2
-1
-2
3
4
5

when n = 5
1
2
-3
-4
-5

Replacing l with X in -->hello world
heXXo worXd

Encoding hello world to hex
68656C6C6F20776F726C64
```

2. Write a class for representing an ASCII picture, **without using the C++ standard library container** and **use the free store memory** to store the data, along with basic operations and test programs.

2.1) Write a `Picture` class which stores a text string of `W×H` characters for its content of which `W` represents the width and `H` represents the height. You are required to:

Provide appropriate **constructors** for class `Picture`

Provide an appropriate **destructor** for class `Picture`

Provide appropriate **copy constructor** for class `Picture`

Provide appropriate **assignment operator** for class `Picture`

Provide appropriate **member functions** for getting the width and the height of a `Picture` object

Provide a **member function**, `print` to print the contents of a picture to the output stream

- Write a test program for testing all use cases of a `Picture` object and its operations including the test for constructing `Picture` object, getting its width and height, printing its contents, passing `Picture` to a function, returning `Picture` from a function, constructing a `Picture` from another `Picture`, and copying a `Picture` object. Verify that you free the memory correctly in the **destructor** of the class.

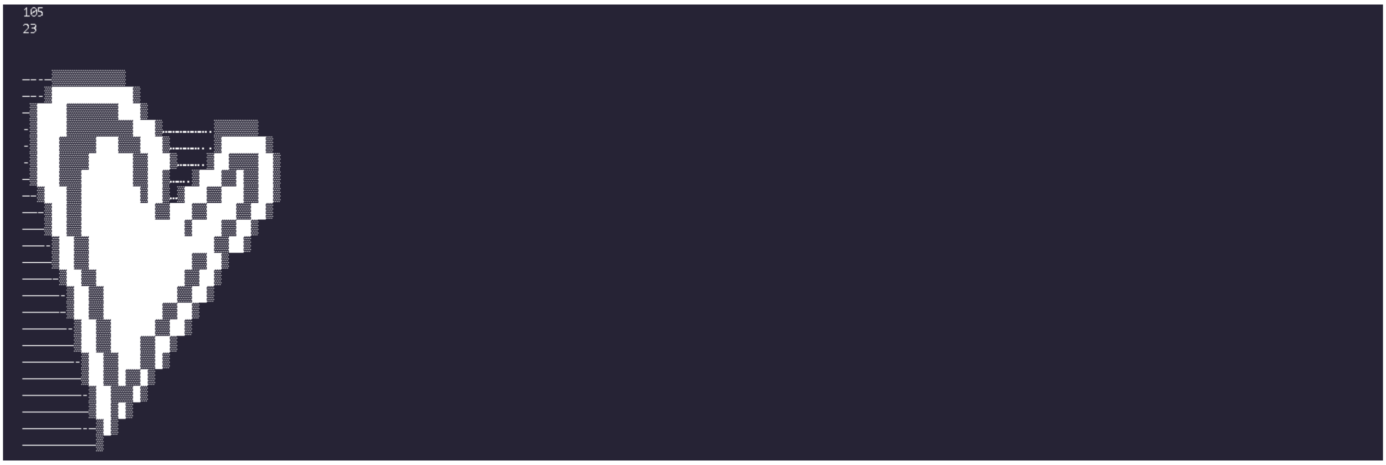
2.2) Modify the code from **2.1)** add the following operations:

- Member function, `clear()`, for deallocating all free store memory used by `Picture` object
  - After calling `pic.clear()` for the `Picture` object `pic`, its width and height should be zero and the object would contains no data for its contents

- **hcat** for creating a new picture by **concatenating** two pictures **horizontally** `vcat` for creating
- **vcat** a new picture by **concatenating** two pictures **vertically**

Add additional support operations as needed. Write a test program for testing all of the above operations.

2.3) Modify the code from **2.2)** by adding a function `resize` to adjust the width and height of a picture. The function will expand the picture size when the new width/height is larger and crop the picture when the new width/height is smaller. Add additional support operations as needed. Finally, write a test program for testing the function.



```
105
23
```