COMP2611 Spring 2021 Homework #3

(Due 11:55PM, Wednesday, 7 April, 2021)

Notes:

- The deadline of this homework is at 11:55pm on Wednesday, 7 April 2021.
- **Write your code in given MIPS assembly skeleton files. Add your own code under TODOs in the skeleton code. Keep other parts of the skeleton code unchanged.**
- Make procedure calls with the registers as specified in the skeleton, otherwise the provided procedures may not work properly. Preserve registers according to the MIPS register convention on slide 76 of the ISA note set.
- Zip the three finished MIPS assembly files into a single zip file, *<your_stu_id>.zip* file (without the brackets). Do not change names of the given skeleton files.
- To submit, first find the Canvas page of COMP2611, homework3, and then upload the "*<your_stu_id>.zip*" file. You can upload for multiple times, only the most updated one before the deadline will be marked.
- Solutions of this homework will be posted at the course web right after the deadline, so **no late submission will be accepted**.
- **Your submitted program must be able to run under MARS, otherwise it will not be marked.**

Question 1: RGB to Grayscale Conversion (20 marks)

The following C++ program gets the Red, Green and Blue (RGB) components of a color as inputs and converts them to the corresponding grayscale of the color.

```
Grayscale = ((R*19595 + G*38469 + B*7472) >>16)
```

The RGB components are unsigned integers between 0 and 255 and are stored in the array `rgb`. The operation “`>>16`” is shift right logical for 16 bits.

C++ Program for RGB to Grayscale Conversion

```cpp
#include <iostream>
using namespace std;
int multiply(int a, int b)
{ // return a*b
    int sum = 0;
    for (int i = 0; i < b; i++)
        sum += a;
    return sum;
}

int RGB2Gray(int rgb[])
{
    // A fast approximation from RGB to grayscale
    int gray = (multiply(rgb[0], 19595) +
                multiply(rgb[1], 38469) +
                multiply(rgb[2], 7472)) >>
               16;
    return gray;
}

int main()
{
    int rgb[3];
    cout << "Input R, G, B value of a point!" << endl;
    cout << "Red (0-255): " << endl;
    cin >> rgb[0];
    cout << "Green (0-255): " << endl;
    cin >> rgb[1];
    cout << "Blue (0-255): " << endl;
    cin >> rgb[2];
    int gray = RGB2Gray(rgb); // convert RGB into grayscale value
    cout << "The grayscale is: " << gray << endl;
    return 0;
}
```

Q1.asm is the MIPS implementation of RGB to Grayscale algorithm. Your job is to implement the `rgb2gray` procedure in MIPS under the TODO, so that it works similarly as the `RGB2Gray()` function of the C++ program. **Add your own code under TODOs in the skeleton code. Keep other parts of the skeleton code unchanged.**

For simplicity, the `multiply` procedure is given. You can always assume the user input is a positive integer within [0, 255].

A sample run of the MIPS program:

```
Input R, G, B value of a point!
Red (0-255):
142
Green (0-255):
229
Blue (0-255):
238
The grayscale is: 204
```

Question 2: Palindrome (20 marks)

A palindrome is a sequence of characters (i.e. string) which reads the same backward as forward, such as `madam` or `racecar`.

The following C++ program takes an English word stored in character array `str[]`and checks whether it is palindrome or not. For simplicity, you may assume the English word contains only valid lower-case English letters, no white spaces, and no punctuation marks. It can always be saved in `str[]`.

C++ Program for Palindrome Check

```cpp
#include <iostream>
using namespace std;

bool isPalindrome(char str[], int n)
{
    for (int i = 0; i < n / 2; i++)
    {
        // compare the two ends of the string
        if (str[i] == str[n - 1 - i])
            continue; // if they are the same, check the next pair
        else
            return false; // if they are different, return false
    }
    return true;
}

int main()
{
    char str[51];

    cout << "Type the string (with length at most 50): " << endl;
    cin >> str;

    int length = 0;
    while (str[length] != 0)
        length++

    cout << boolalpha << isPalindrome(str, length) << endl;

    return 0;
}
```

Q2.asm is the MIPS implementation of palindrome checking. Your job is to implement the `strlen` and `isPalindrome` procedures in MIPS under the TODO, so that they work similarly as the related functions in the C++ program. **Add your own code under TODOs in the skeleton code. Keep other parts of the skeleton code unchanged.**

Two sample runs of the MIPS program:

```
Type the string (with length at most 50):
neveroddoreven
true

-- program is finished running –

Type the string (with length at most 50):
itsnotapalindrome
false
-- program is finished running –
```

Question 3: Find and Replace All Occurrences of a Substring (20 marks)

The following C++ program replaces all occurrences of a substring with another substring of the same length.
Assume the original string is saved in `str[]`, its substring (to be replaced) is stored in `substr1[]`, and the replacing substring is stored in `substr2[]`.

Function `FindAndReplaceSubstring()` works as following, let's assume we start with `i=0`:

1. Character `str[i]` in the original string will be checked to see if it matches with the first character of substring1 `substr1[0]`.
2. If the character `str[i]` is the same as `substr1[0]`, a loop is launched to check whether the subsequent characters in `str` are exactly the same as the remaining characters in `substr1`.
3. If a matching can't be found in step 2, return to step 1 to start searching from the next character of the original `str` (`i=i+1`). If a matching is found in step 2, the function `Replace()` will be executed to replace the current occurrence of `substr1` with `substr2` in the original string `str`. Then we will start from a position right after the replacement in `str` (`i=i+substrlength-1`) and repeat from step 1.
4. The search loop (steps 1-3) will continue running until all the characters of the original string `str` are checked with `substr1[0]`. See the C++ function `FindAndReplaceSubstring()` for the details.

Your job is to refer to the C++ program and implement MIPS procedures `findAndReplaceSubstring` and `replace` in the *Q3.asm*. The two MIPS procedures should work like the corresponding functions in C++ program.

Note:

- For simplicity, you can assume the user always inputs valid string which fits in the character array.
- You can re-use the code to calculate string length in Question 2.

Add your own code under TODOs in the skeleton code. Keep other parts of the skeleton code unchanged.

C++ Program for Substring Replacement

```
#include <iostream>
#include <cstring>
using namespace std;

void Replace(char str[], int index, char substr2[], int substrLength) {
    for (int i=0; i<substrLength; i++) {
        // replace the substring1 starting at index with the substring2
        str[index+i] = substr2[i];
    }
    return;
}

void FindAndReplaceSubstring(char str[], char substr1[], char substr2[]
) {
    bool processed = false;
    int length, substrlength;

    length=strlen(str);
    substrlength=strlen(substr1);

    for (int i=0; i<length; i++) {
        if(str[i]==substr1[0]) { // step 1
            for (int j=0; j<substrlength; j++) { // step 2
                if (str[i+j] != substr1[j]) { // step 3: no match
                    break;
                }
                if (j == substrlength-1) { // step 3: match and replace
                    Replace(str, i, substr2, substrlength);
                    i=i+substrlength-1;
                    processed = true;

                }
            }
        }
    }
    if (processed)
        cout << "The processed string: " << str << endl;
    else
        cout << "substring not found!" << endl;
}

int main() {
    char str[50];
    char substr1[50];
    char substr2[50];

    cout << "Input the string (with length at most 50): ";
    cin >> str;
    // assume substr1, substr2 entered by the user are always of
    // exactly the same length
    cout << "Input the substring you want to replace: ";
    cin >> substr1;
    cout << "Input the substring you want to replace with (in the same
length): ";
    cin >> substr2;
    FindAndReplaceSubstring(str, substr1, substr2);
}
```

Two sample runs of the MIPS program:

```
Input the string (with length at most 50):
ssubsubsubs
Input the substring you want to replace:
sub
Input the substring you want to replace with (in the same length):
abc

The processed string is :sabcabcabcs

-- program is finished running –

Input the string (with length at most 50):
abaabaaabcdabb
Input the substring you want to replace:
de
Input the substring you want to replace with (in the same length):
hi
substring not found!
-- program is finished running --
```