Homework 4
CDA 3104 Computer Organization and Assembly Language Programming
Due date: Nov. 22, 2012 11:59pm

Requirements:

1. This assignment as well as other assignments in this class must be finished on Windows operating system.
2. Zip your program and submit the zip file on ANGEL

Assignments:

It is a common programming practice to search for a character in a string. The most intuitive method is to do a linear search, while it is not the most efficient. Say the length of the string is m and you have n characters to search for, you will have to make ***m*n*** comparisons in the worst case.

In this assignment, you are going to implement an algorithm whose performance is ***m+n***. Below is the pseudo code:

```
// building a map that records the existence of characters in a string
byte map[32]={0,0……0};
int len=strlen(string);
for(int i=0;i<len;i++)
      map[(string[i])>>3] |=  1<<( (string[i]) &7)

//deciding whether a character (ch) is in string
if ( (map[ch>>3] & (1<<(ch&7))) !=0 )
      //ch is in string
else
      //ch is not in string
```

In the above program, some bit operators are used. Here are their meanings:

op1 >> op2: shift the bits in op1 to the right for op2 bits; neither op1 nor op2 is affected

op1 << op2: shift the bits in op1 to the left for op2 bits; neither op1 nor op2 is affected

op1 & op2: perform bit-wise AND operation between each pair of matching bits in op1 and op2; neither op1 nor op2 is affected

op1 | op2: perform bit-wise OR operation between each pair of matching bits in op1 and op2; neither op1 nor op2 is affected

In your program, please implement two procedures: PrepareMap and CharacterSearch. Their prototypes are listed as follows:

```
; For each character in string, register it in map
PrepareMap PROTO,
      string: PTR BYTE,
      map: PTR BYTE
```

```
; Set ZF if char does NOT exist in the string;
; Clear ZF otherwise.
CharacterSearch PROTO,
      map: PTR BYTE,
      char: BYTE
```

Additionally, please use the following data segment without adding any new variables.

```
.data
stringMap byte 32 dup(0)
testString byte "hello, assembly language programming!",0

msg1 byte " is not in the string.",0
msg2 byte " is in the string.",0
```

You are also supposed to write a main procedure for testing. In the main procedure, please randomly generate 10 lower case letters and for each letter use your CharacterSearch procedure to decide if it is in testString.

Hints:

1. "invoke str_length, aStringOffset" is a procedure in the textbook library and returns the length of a string to EAX.
2. It is easy for you to decide to use the LOOP instruction when implementing PrepareMap. However, you need to be concerned when using it. Why?

Grading Policies:

Correctly implement a testing driver (main procedure)	20%
Successfully implement procedure `PrepareMap`	40%
Successfully implement procedure `CharacterSearch`	30%
Correct format of displaying	10%