

Homework 2 - Lisp



Installation

SBCL

Steel Bank Common LISP

A high performance **ANSI Common Lisp** implementation.

SBCL Official Website:
<http://www.sbcl.org/index.html>

Install SBCL on Windows

Go to this website,

<https://github.com/akovalenko/sbcl-win32-threads/wiki>

32 bit:

[sbcl-1.1.4.0.mswin.1288-90ab477-x86.msi](#)

64 bit:

[sbcl-1.1.4.0.mswin.1288-90ab477-x86-64.msi](#)

Install SBCL on Linux

use **APT**

```
$ sudo apt-get install sbcl
```

use **YUM**

```
$ sudo yum install sbcl
```

others

```
That's your business...XD
```

Install SBCL on OS X

use **homebrew**

```
$ brew install sbcl
```

use **macports**

```
$ sudo port install sbcl
```

If you never believe anyone,

Compile by yourself...

<http://www.sbcl.org/getting.html>

Command Line Interface

for Windows

命名提示字元

for Unix-like OS

終端機

Interactive Env

```
$ sbcl
```

Interactive Env

```
$ sbcl
```

```
This is SBCL 1.0.57.0.debian, an implementation of ANSI  
Common Lisp.
```

```
More information about SBCL is available at  
<http://www.sbcl.org/>.
```

```
SBCL is free software, provided as is, with absolutely no  
warranty.
```

```
It is mostly in the public domain; some portions are  
provided under  
BSD-style licenses. See the CREDITS and COPYING files in  
the  
distribution for more information.
```

```
*
```

* (+ 1 2)

* (+ 1 2)

3

* (+ 1 2)

3

* (+ 1 2 3)

* (+ 1 2)

3

* (+ 1 2 3)

6

```
* ( + 1 2 )
```

```
3
```

```
* ( + 1 2 3 )
```

```
6
```

```
* "Hello World"
```

```
* ( + 1 2 )
```

```
3
```

```
* ( + 1 2 3 )
```

```
6
```

```
* "Hello World"
```

```
"Hello World"
```



```
* ( + 1 2 )
```

```
3
```

```
* ( + 1 2 3 )
```

```
6
```

```
* "Hello World"
```

```
"Hello World"
```

```
* (DEFUN FIB(n)
  (IF (< n 2)
    n
    (+ (FIB (- n 1)) (FIB (- n 2))))))
```

```
* ( + 1 2 )
```

```
3
```

```
* ( + 1 2 3 )
```

```
6
```

```
* "Hello World"
```

```
"Hello World"
```

```
* (DEFUN FIB(n)
  (IF (< n 2)
    n
    (+ (FIB (- n 1)) (FIB (- n 2))))))
```

```
FIB
```

```
* ( + 1 2 )
```

```
3
```

```
* ( + 1 2 3 )
```

```
6
```

```
* "Hello World"
```

```
"Hello World"
```

```
* (DEFUN FIB(n)
  (IF (< n 2)
    n
    (+ (FIB (- n 1)) (FIB (- n 2))))))
```

```
FIB
```

```
* (FIB 20)
```

```
* ( + 1 2 )
```

```
3
```

```
* ( + 1 2 3 )
```

```
6
```

```
* "Hello World"
```

```
"Hello World"
```

```
* (DEFUN FIB(n)
  (IF (< n 2)
    n
    (+ (FIB (- n 1)) (FIB (- n 2))))))
```

```
FIB
```

```
* (FIB 20)
```

```
6765
```

```
* ( + 1 2 )
```

```
3
```

```
* ( + 1 2 3 )
```

```
6
```

```
* "Hello World"
```

```
"Hello World"
```

```
* (DEFUN FIB(n)
  (IF (< n 2)
    n
    (+ (FIB (- n 1)) (FIB (- n 2))))))
```

```
FIB
```

```
* (FIB 20)
```

```
6765
```

```
* (exit)
```

```
* ( + 1 2 )
```

```
3
```

```
* ( + 1 2 3 )
```

```
6
```

```
* "Hello World"
```

```
"Hello World"
```

```
* (DEFUN FIB(n)
  (IF (< n 2)
    n
    (+ (FIB (- n 1)) (FIB (- n 2))))))
```

```
FIB
```

```
* (FIB 20)
```

```
6765
```

```
* (exit)
```

```
$
```

Script File

```
1      ;;; file: fib.lsp
2
3      (DEFUN FIB(n)
4        (IF (< n 2)
5          n
6          (+ (FIB (- n 1)) (FIB (- n 2)))))
7      )
8    )
9
10     (print (FIB 20))
11
12
13
14
```

Execution

```
$ sbcl --script fib.lsp
```

```
6765
```


If you still have any question about SBCL,
Read The Friendly Manual.

ANSI Common Lisp Tutorial:

<http://acl.readthedocs.org/en/latest/>

SBCL Manual:

<http://www.sbcl.org/manual/index.html>

**Read The Friendly Manual or
Use The Friendly Google first,**
before you ask teacher or TAs.

可自行選題完成此次作業:

Problem 1.1~1.3: 滿分 80分

Problem 2 (Merge Sort): 滿分90

Problem 3 (Diff Command): 滿分100分

Problem 1.1

- Write a Common Lisp function that checks whether the input number is a prime number. (Hint: 2 is a prime number. You may call a function to do the checking loop)
- Test data : (function name : prime)
- (prime 2) (prime 239) (prime 999) (prime 17)

Problem 1.2

- Write a Common Lisp function that takes a simple list and determine whether its contents form a palindrome. (hint: there are many different ways to do it. One of them is to define reverse and equal functions.)
- test data : (**function name : palindrome**)
- (palindrome '(a b c))
- (palindrome '(m a d a m))
- (palindrome '(cat dog))
- (palindrome '())
- (palindrome '(cat dog bird bird dog cat))

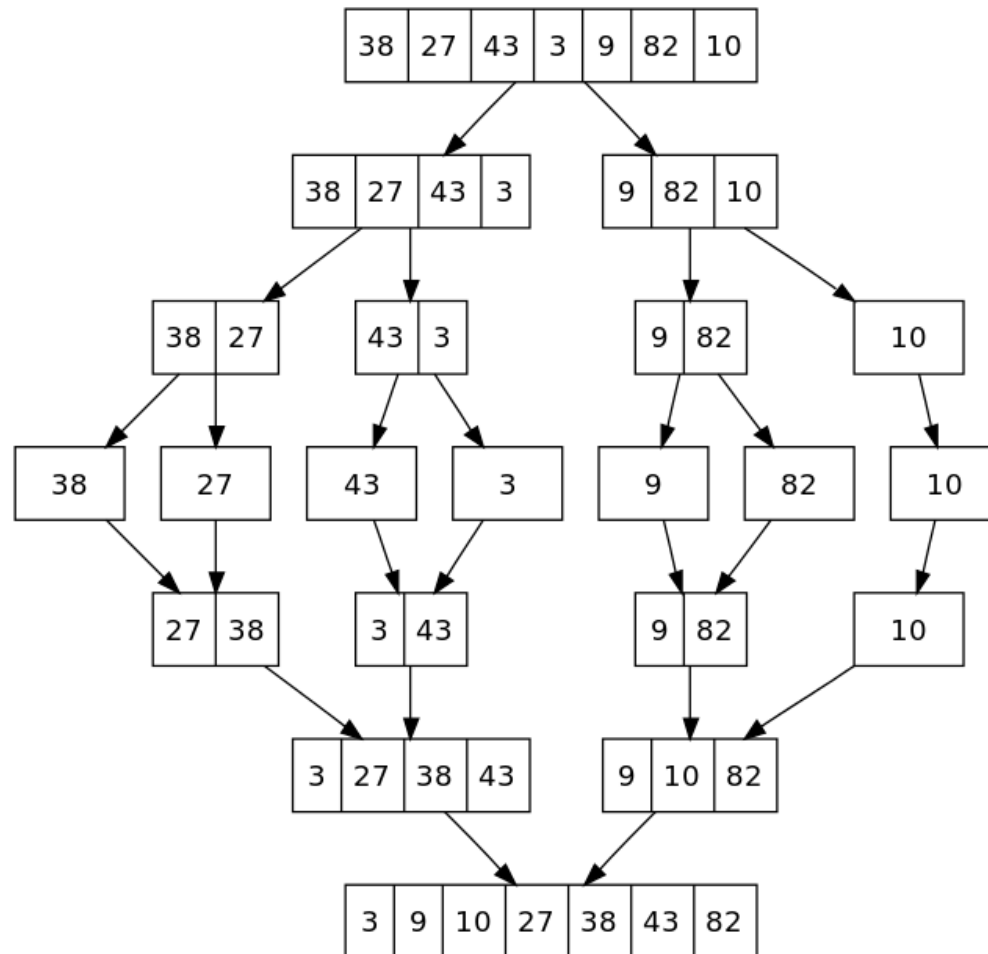
Problem 1.3

- Write a Common Lisp function about Fibonacci function with **original recursion** and **tail recursion**.
- Original func. name : **fib1** Tail func. name : **fib2**
- Use trace or dtrace to show the execution details :

```
Break 2 [71]> <trace fib1>
;; Tracing function FIB1.
<FIB1>
Break 2 [71]> <fib1 3>
2. Trace: <FIB1 '3>
3. Trace: <FIB1 '2>
4. Trace: <FIB1 '1>
4. Trace: FIB1 ==> 1
4. Trace: <FIB1 '0>
4. Trace: FIB1 ==> 0
3. Trace: FIB1 ==> 1
3. Trace: <FIB1 '1>
3. Trace: FIB1 ==> 1
2. Trace: FIB1 ==> 2
2
```

```
Break 3 [81]> <trace fib2>
;; Tracing function FIB2.
<FIB2>
Break 3 [81]> <fib2 8>
3. Trace: <FIB2 '8>
3. Trace: FIB2 ==> 21
21
```

Problem 2: Merge Sort



輸入格式

- 先輸入一個整數 N 表示數列中有多少數字
- 再輸入 N 個整數作為數列中的數字

範例輸入

Example 1:

3

3 2 1

Example 2:

5

1 3 8 9 1

Example 3:

10

9 8 16 2 7 199 0 98 1 29

輸出格式

- 輸出包含 N 個整數的已排序數列

範例輸出

Example 1:

1 2 3

Example 2:

1 1 3 8 9

Example 3:

0 1 2 7 8 9 16 29 98 199

參考程式碼

可基於以下程式碼進行修改

```
(defun mergesort (numbers)
  (return-from mergesort numbers))

; main function
(let ((n (read))
      (numbers))
  (setf numbers
    (do ((i 0 (+ i 1))
        (tmp nil))
      ((>= i n)
       (reverse tmp))
      (setf tmp (cons (read) tmp))))
  (format t "~{~A ~}~%" (mergesort numbers)))
```

Problem 3: `diff` command

`diff` is an useful command on Unix-based system. You can use it to compare two files. It will show you the difference between them.

diff command

hello-world.c

```
#include <stdio.h>
int main() {
    printf("Hello World");
    return 0;
}
```

hello-world.cpp

```
#include <iostream>
using namespace std;
int main() {
    cout << "Hello World" << end;
    return 0;
}
```

diff command

```
$ diff hello-world.c hello-world.cpp
--- hello-world.c      2016-04-10 19:22:38.000000000 +0800
+++ hello-world.cpp    2016-04-10 19:22:45.000000000 +0800
@@ -1,6 +1,7 @@
-#include <stdio.h>
+#include <iostream>
+using namespace std;
 int main() {
- printf("Hello World");
+ cout << "Hello World" << end;
 return 0;
 }
```


Input

- The filenames of two input files must be `file1.txt` and `file2.txt`.

Output

- Output the result like `"$ diff file1.txt file2.txt"`.
- Output the content part only. It means we don't need some parts like `"@@ -1,6 +1,7 @@"`.
- Keep all the same lines in the output.
- No need to colorize the output.
 - If you want, you can do it.

Sample Output

```
$ sbcl --script diff.lsp
-#include <stdio.h>
+#include <iostream>
+using namespace std;
  int main() {
-   printf("Hello World");
+   cout << "Hello World" << end;
    return 0;
  }
```

Sample Output

```
$ sbcl --script diff.lsp
```

```
-#include <stdio.h>
```

```
+#include <iostream>
```

diff chunk #1

```
+using namespace std;
```

```
int main() {
```

```
- printf("Hello World");
```

```
+ cout << "Hello World" << end;
```

diff chunk #2

```
return 0;
```

```
}
```

Sample Output

```
$ sbcl --script diff.lsp
-#include <stdio.h>
+#include <iostream>
+using namespace std;
int main() { same lines #1
- printf("Hello World");
+ cout << "Hello World" << end;
return 0;
} same lines #2
```

Deadline

2018/04/18 22:00