Homework V

Policies

- This homework is due Monday, 11/10/2014, 00:01 am.
- Late hand-ins will be accepted until the same day, 08:00 am. They will be penalized with 10% of the maximum achievable score.
- If you have any questions, you can use the Piazza forum for our class: https://piazza.com/uci/fall2014/141/home - please make sure to select the correct folder. Do not use Piazza to post any partial solutions.
- Include your name and your student id on all pages and in all source code files.
- The submission process is described below.

Exercise 1 (10 Points)

Write a first-fit allocator using the alloc.c template given on EEE. The positions that you need to modify are marked with "TODO". We provided print methods that show you the current state of the arena once you have implemented createArena. It helps to start by understanding how the print methods extract information from the buffer and how they cast information from the generic arena memory into slot entries.

See the last page for an example on how the allocator should behave. "M" marks meta data entries, "X" marks occupied slots and " " marks free slots.

Here is a resource that can be helpful in understanding the first-fit algorithm: http://www.inf.udec.cl/~leo/Malloc_tutorial.pdf

Given the issues in previous homework assignments where compilers generated programs from incorrect code that worked while others failed to do so, **please make sure that your program compiles and works with gcc 4.2 under Linux**. You are still free to use whatever IDE and compiler you want, but the grading will use this configuration. If you do not use Linux, you can use the ICS OpenLab SSH Cluster: https://www.ics.uci.edu/computing/linux/hosts.php

Exercise 2 (5 Points)

Implement the following prolog relations. If applicable, the relation should be able to respond to the backward query. You may use built-in predicates append, member, not, reverse, and length.

Define the predicate compute-change(Money,Quarter,Dime,Nickle,Penny). You may assume that Money is less than 100 (cents) and you want to minimize the number of coins you should give. (No backward query)

```
compute-change(33,Q,D,N,P) -> Q = 1 D = 0 N = 1 P = 3
```

Exercise 3 (5 Points)

Define the predicate palindrome(Base,Result). You may assume that Result is always even length list. However, you must show that your answer is the only possible answer.

```
palindrome([m,a,d],R).
     R = [m,a,d,d,a,m] ;
     no
palindrome(B,[n,i,s,s,i,n]).
     B = [n,i,s] ;
     no
```

Submission

Please submit your answers to the EEE Dropbox for this homework (e.g. COMPSCI 141 HW 5). You can find instructions on how to do this here:
https://eee.uci.edu/help/dropbox/students/#studentsubmit

Submit a zip file with the following contents:

- **alloc.c**
- **prolog.pl**

Reminder: .doc files are not .pdf files, .rar files are not .zip files.

In addition to reading your code, we will submit it to an automated test suite. For this reason, it is important that your submission follows this format. Make sure your code compiles without errors.

```
BEGIN
[M                                                           ]

ALLOC a
[MXM                                                         ]
  ^

ALLOC b
[MXMXXM                                                      ]
    ^

ALLOC c
[MXMXXMXXXXXM                                                ]
       ^

DEALLOC a
[M MXXMXXXXXM                                                ]

DEALLOC c
[M MXXM                                                      ]

ALLOC d
[MXMXXM                                                      ]
  ^

DEALLOC b
[MXM                                                         ]

ALLOC e
[MXMXM                                                       ]
    ^

DEALLOC e
[MXM                                                         ]

ALLOC f
[MXMXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXM             ]
    ^

ALLOC g
[MXMXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXMXXM          ]
                                                ^

ALLOC h
[MXMXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXMXXMXXXXXM    ]
                                                   ^

ALLOC i
[MXMXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXMXXMXXXXXMXXXX]
                                                         ^

DEALLOC d
[M MXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXMXXMXXXXXMXXXX]

DEALLOC f
[M                                             MXXMXXXXXMXXXX]

DEALLOC g
[M                                                MXXXXXMXXXX]

DEALLOC h
[M                                                      MXXXX]

DEALLOC i
[M                                                           ]
```