

Concordia University
Department of Computer Science and Software
Engineering
SOEN 6411-AA:
**Comparative Study of Programming
Languages**

Assignment 3 on C

Dr. Constantinos Constantinides, P.Eng.
`constantinos.constantinides@concordia.ca`

June 15, 2021

1 General information

Date posted: Tuesday, 15 June, 2021.

Date due: Friday 25 June, 2021, by 23:59.

Weight: 30% of the overall mark.

2 Introduction and ground rules

1. This is a team assignment. Each team should designate a leader who will submit the assignment electronically.
2. You may not seek any assistance while expecting to receive credit. You must work strictly within your team and seek no assistance for this project (from the instructor, the teaching assistants, fellow classmates and other teams or external help). Failure to do so will result in penalties or no credit.
3. All team members are expected to work relatively equally on each Problem. The team leader has the responsibility to ensure that the team does not violate this rule. Failure to do so will result in penalties. In your submission, you must include only the names of those people who contributed to the assignment. Accommodating someone who did not contribute will result in penalties to each team member and a heavier penalty for the team leader.
4. If there is any problem in the team (such as lack of contribution, etc.), the team leader must contact the instructor as soon as the problem appears.
5. No late submissions will be accepted.

3 Your assignment

In this exercise we want to simulate the behavior of LISP's list construction in C, where the elements are limited to lists and atoms of “char” type only.

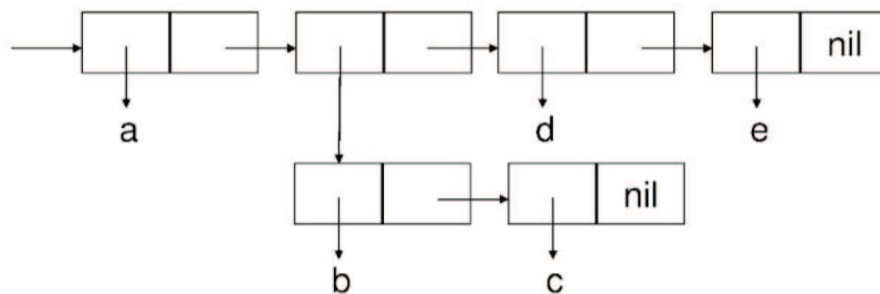
```
typedef enum { ATOM, LIST } eltype;
typedef char atom;
struct _listnode;
typedef struct {
    eltype type;
    union {
        atom a;
        struct _listnode* l;
    };
} element;
typedef struct _listnode {
    element el;
    struct _listnode* next;
} * list;
const element NIL = { .type=LIST, .l=NULL };
```

Using the above definition, implement the following functions:

1. `element aasel(atom a)`; AKA atom as element, returns an element whose content is set to atom *a*.
2. `element lasel(list l)`; AKA list as element, returns an element whose content is set to the list, pointed by *l*.
3. `list cons(element e, list l)`; that creates a new list whose *car* and *cdr* are the element *e* and the list *l*. While the memory for the newly created list is to be allocated dynamically.

4. `list append(LIST l1, list l2)`; that creates a new list whose elements are shallow copies of elements in *l1* and *l2*, appended.
5. `element car(element e)`; that returns head of the list, represented by *e*; returns NIL, if *e* is not a list.
6. `list cdr(element e)`; that returns tail of the list, represented by *e*.
7. `list cddr(element e)`; that similarly returns the `cddr` of the list, represented by *e*.
8. `void print(e)`; that prints the content of the element *e*. If *e* is an atom, it prints the symbol enclosed in spaces, and if *e* it is a list, it prints recursively prints the elements of the list enclosed in parentheses. If *e* is NIL, the word “NIL” must be printed.
9. `void free(LIST l)`; that frees all the memory previously allocated by the whole list (including all its elements and its inner lists)

Write a short code to create and display the following list:



Additionally, print the `car` and the `cdr` of the above list; also print the `car` of the `car` of the original list.

The output must look like the following:

```
( a ( b c ) d e )
```

```
a
```

```
(( b c ) d e )
```

```
NIL
```

Make sure the list is freed before your program terminates.

4 What to submit

You must submit a zip file containing the following two files:

1. Your source code.
2. A README.txt file with all names that contributed to the assignment.

Name the zip file after your team e.g. `team1.zip` and submit it at the Electronic Assignment Submission portal

(<https://fis.encs.concordia.ca/eas>)

under **Programming Assignment 3**.

END OF ASSIGNMENT
