
Linked Lists and ADT

CS19001: Programming and Data Structures Laboratory

05-Nov-2018



Soumyajit Dey and Aritra Hazra

Department of Computer Science & Engineering,
Indian Institute of Technology Kharagpur,
Paschim Medinipur, West Bengal, India – 721302.

http://cse.iitkgp.ac.in/~aritrah/course/lab/PDS/Autumn2018/CS19101_PDS-Lab_Autumn2018.html

Programming Assignments

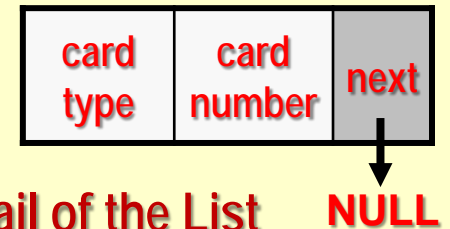
Complete and submit during lab

Assignment 1 [Card-Matching Game]

- ❑ Suit / Card Nomenclature (Suit = 4 Decks of 13 Cards Each)
 - Card Numbers (Ascending Order of Value): 2–10, J, Q, K, A
 $2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < J \text{ (Jack)} < Q \text{ (Queen)} < K \text{ (King)} < A \text{ (Ace)}$
 - Deck Types: Spades (♠/S), Hearts (♥/H), Diamonds (♦/D) and Clubs (♣/C)
- ❑ Write a C-program to simulate the following 2-player game-of-cards:

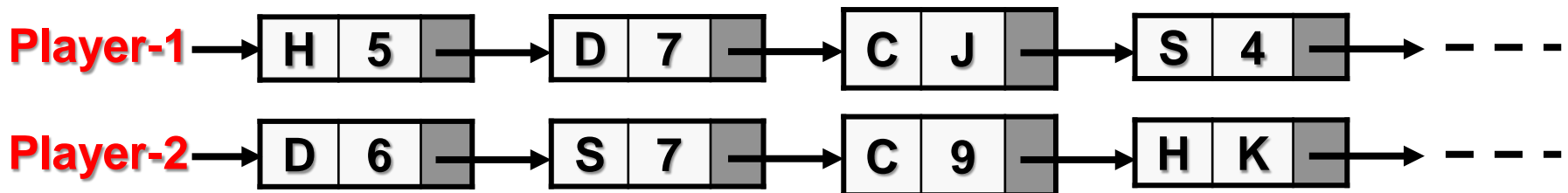
- **Task-1:** Write few basic Linked List functions/code

- Define a node in the list as shown in the right
- Write a function to Create a New Node
- Write a function to Insert a New Node at the End/Tail of the List
- Write a function to Delete the Head/Front Node from the List



- **Task-2:** Write a Function to –

- Distribute total 52 cards *randomly* into two hands of 26 unique cards each
- Keep these cards in two separate linked list structure for both players



Assignment 1 [Card-Matching Game]

■ Task-3: Write C main-function for the game (goes in rounds) as follows:

Task-3a:

Write a C-function to draw (in each round) two cards (by deleting two players' head/front nodes of list).

Task-3b:

Write a C-function to match two drawn cards (i.e. two nodes) and determine the larger or equal card number.

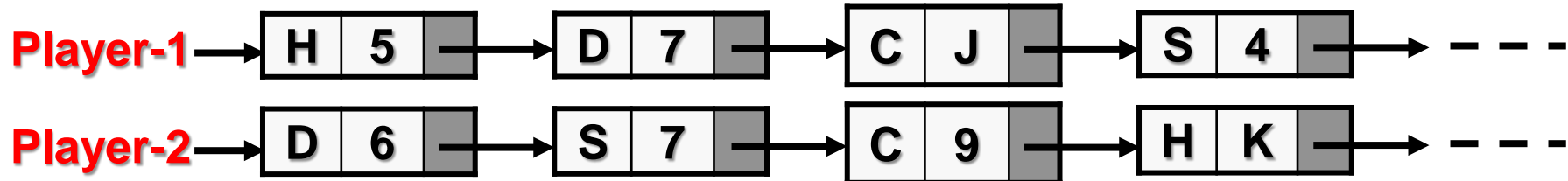
- Initially, Player-1 will start. From next round, the first turn will be from the player who wins the last round.
- In each round, both player takes out their first card (following the order of turns) from the beginning of their respective hands and put this pair one over the other.
- If the numbers printed on the **two drawn cards are SAME**, the round continues, and next cards are taken out from both players (turn-wise) ... so on ...
- A player **WINS a round** when the number printed on his/her card is **MORE** than his/her opponent player in the last pair of cards drawn. In such case, all the standing cards (of that given round) are acquired by the winning player and (s)he puts the cards (in same order as drawn) at the end of his/her hand of cards.
- ... Similarly, the next round repeats the same play ... on and on ...
- **Finally**, if a player can acquire all 52 cards, then (s)he is declared as the **WINNER!** If both of the players hand finishes together[#], then the match is declared as **TIED!**

Task-3c: Write a C-function to build a list (in each round) of drawn cards turn-wise and append into the end of the player list (in order of their play-turns) whoever wins that round.

[#]It may happen rarely – only when the sequence of cards follow same numbering in both hands

Assignment 1 : *Simulation for Game-of-Cards*

Initial Hands:



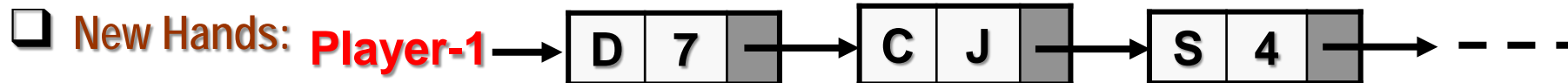
Play #1 (Turn of Player-1 first):

H	5
---	---

 vs

D	6
---	---

 → Player-2 takes ...



Play #2 (Turn of Player-2 first):

D	7
---	---

 vs

S	7
---	---

 → Same! Next Turn ...

Play #3 (Turn of Player-2 first):

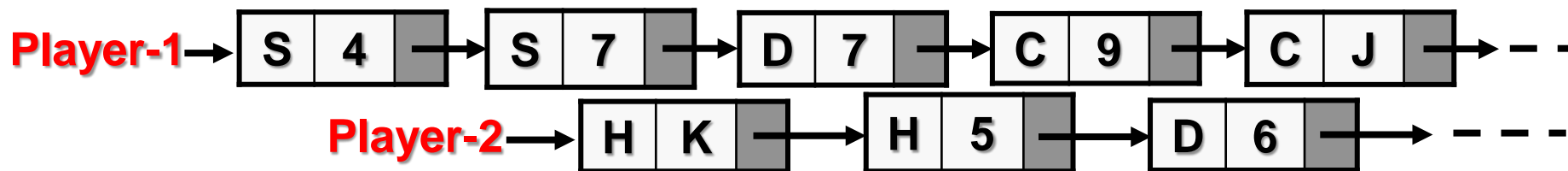
C	J
---	---

 vs

C	9
---	---

 → Player-1 takes ...

New Hands:



Assignment 1: *Solution Aids*

❑ Random Number Generation

```
#include <stdlib.h>           // for srand(), rand()
#include <sys/types.h>        // for getpid()
#include <unistd.h>           // for getpid()

int main()
{
    // declare srand() in the beginning of main (once)
    srand( getpid() );

    // generate any random number [0-9]
    x = rand() % 10;
    return 0;
}
```

Thank You!