COS10007

Week 5 Prac

Question 1

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

struct LL {

char character;

struct LL \* next;

};

typedef struct LL ll\_struct;

typedef ll\_struct \* ll;

void cleanup(ll \* freethis);

void print\_list(ll \* printthis);

int main(void) {

// Setup random number generation.

srandom(time(NULL));

// Create the linked list to work on.

ll list0 = NULL;

// Dump random values into the list.

for(int i = 0; i < 10; i++) {

ll newll = (ll\_struct\*) malloc(sizeof(ll\_struct));

newll->character = random() % 26 + 97;

newll->next = list0;

list0 = newll;

}

print\_list(&list0);

// Reverse the order of the list.

ll prev = NULL;

ll next = list0;

while(next != NULL) {

next = list0->next;

list0->next = prev;

prev = list0;

if(next != NULL) {list0 = next;}

}

print\_list(&list0);

// Clean up memory.

cleanup(&list0);

return 0;

}

// Collapse and free ll \*arg1.

void cleanup(ll \* freethis) {

while(\*freethis != NULL) {

ll temp = \*freethis;

\*freethis = (\*freethis)->next;

free(temp);

}

}

// Print contents of ll \*arg1.

void print\_list(ll \* printthis) {

// Print linked list contents onto screen.

for(ll ll\_pos = \*printthis; ll\_pos != NULL; ll\_pos = ll\_pos->next) {

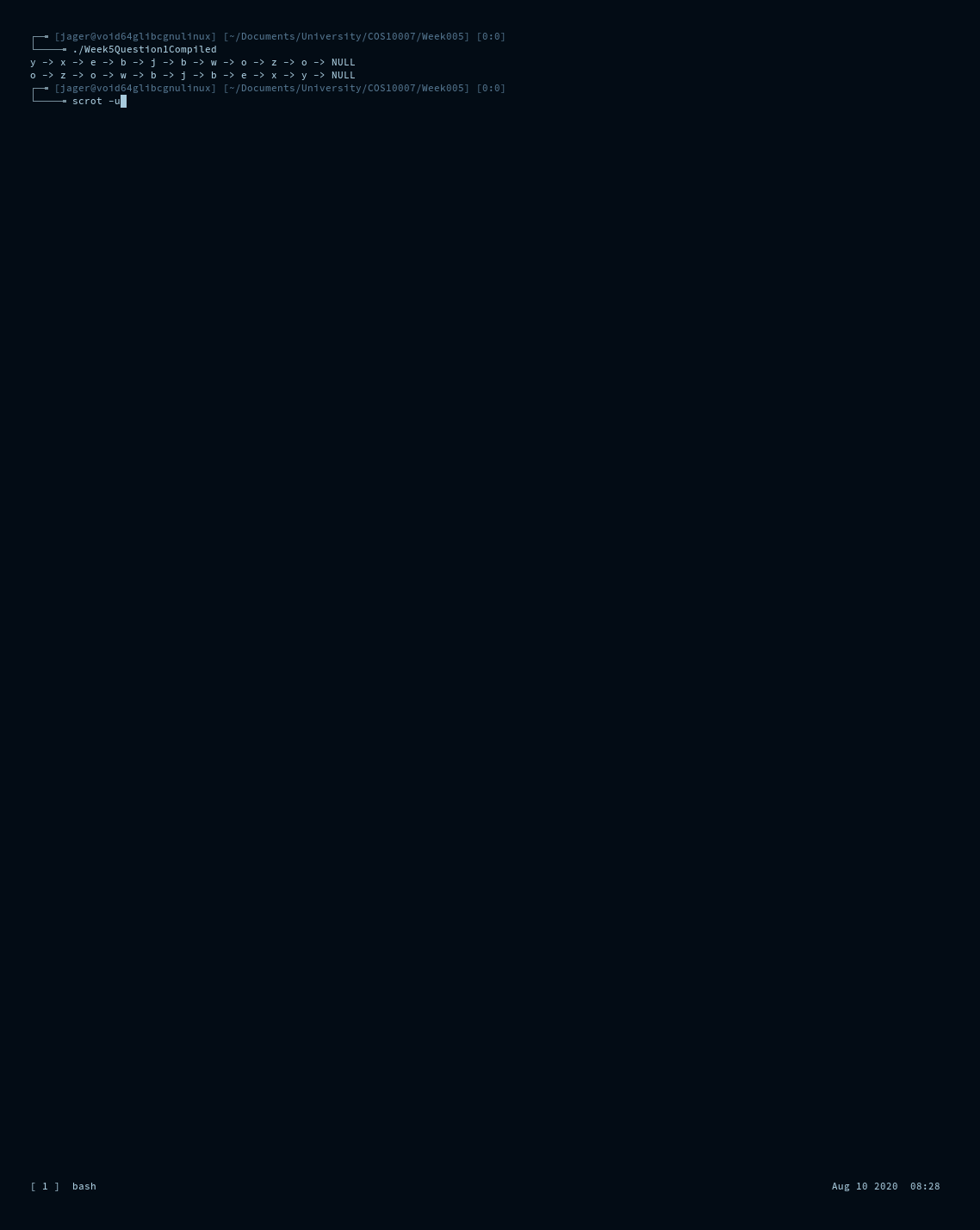
printf("%c", ll\_pos->character);

printf(" -> ");

}

printf("NULL\n");

}



Question 2

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <stdbool.h>

struct LL {

int integer;

struct LL \* next;

};

typedef struct LL ll\_struct;

typedef ll\_struct \* ll;

void cleanup(ll \* freethis);

void print\_list(ll \* printthis);

void print\_list\_in\_reverse(ll \* printthis);

int main(void) {

// Create the linked list to work on.

ll list0 = NULL;

// Dump random values into the list.

for(int i = 0; i < 10; i++) {

ll newll = (ll\_struct\*) malloc(sizeof(ll\_struct));

newll->integer = i;

newll->next = list0;

list0 = newll;

}

print\_list(&list0);

print\_list\_in\_reverse(&list0);

// Clean up memory.

cleanup(&list0);

return 0;

}

// Collapse and free ll \*arg1.

void cleanup(ll \* freethis) {

while(\*freethis != NULL) {

ll temp = \*freethis;

\*freethis = (\*freethis)->next;

free(temp);

}

}

// Print contents of ll \*arg1.

void print\_list(ll \* printthis) {

// Print linked list contents onto screen.

for(ll ll\_pos = \*printthis; ll\_pos != NULL; ll\_pos = ll\_pos->next) {

printf("%d", ll\_pos->integer);

}

printf("\n");

}

void print\_list\_in\_reverse(ll \* printthis) {

bool printfunc(ll ptr) {

if(ptr->next != NULL) {

printfunc(ptr->next);

}

printf("%d", ptr->integer);

}

printfunc(\*printthis);

printf("\n");

}

