LAB NO 8 Q1.

#include <stdio.h> #include <stdlib.h> #include <string.h> #include <math.h>

#define SIZE 250

typedef enum { NO, YES } BOOL; typedef struct Node {

int value;

struct Node \* next; struct Node \* prev;

} NODE\_t;

typedef NODE\_t \* NODE\_p\_t; NODE\_p\_t createNode () {

NODE\_p\_t temp = (NODE\_p\_t)malloc(sizeof(NODE\_t)); temp->value = 0;

temp->prev = temp; temp->next = temp; return temp;

}

void insert (NODE\_p\_t list, int val) { NODE\_p\_t temp = createNode();

NODE\_p\_t p = list->next; temp->value = val;

temp->next = p; temp->prev = list;

list->next = temp; p->prev = temp;

(list->value)++;

}

void inputLongInteger (NODE\_p\_t li) {

int i;

char \* longint = (char \*)malloc(SIZE \* sizeof(char)); scanf(" %s", longint);

for (i = (int)strlen(longint) - 1; i >= 0; --i)

insert(li, \*(longint + i) - '0');

}

NODE\_p\_t createAndInputLongInteger () { NODE\_p\_t li = createNode(); inputLongInteger(li);

return li;

}

NODE\_p\_t addLongIntegers (NODE\_p\_t lia, NODE\_p\_t lib) { NODE\_p\_t lic = createNode();

int carry = 0, sum = 0; NODE\_p\_t tempa = lia->prev; NODE\_p\_t tempb = lib->prev;

while (tempa != lia && tempb != lib) {

sum = tempa->value + tempb->value + carry; carry = sum / 10;

sum = sum % 10; insert(lic, sum); tempa = tempa->prev; tempb = tempb->prev;

}

while (tempa != lia) {

sum = tempa->value + carry; carry = sum / 10;

sum = sum % 10; insert(lic, sum); tempa = tempa->prev;

}

while (tempb != lib) {

sum = tempb->value + carry; carry = sum / 10;

sum = sum % 10; insert(lic, sum); tempb = tempb->prev;

}

if (carry != 0)

insert(lic, carry);

return lic;

}

void displayLongInteger (NODE\_p\_t longInteger, int maxSize) { int size;

for (size = maxSize; size > longInteger->value; --size) printf(" ");

NODE\_p\_t temp = longInteger->next; while (temp != longInteger) {

printf("%d ", temp->value); temp = temp->next;

}

}

int max (int a, int b, int c) {

return (a>b?((a>c)?a:c):(b>c)?b:c);

}

int main (int argc, const char \* argv []) { printf("\n\tEnter A: ");

NODE\_p\_t lia = createAndInputLongInteger(); printf("\n\tEnter B: ");

NODE\_p\_t lib = createAndInputLongInteger(); NODE\_p\_t lic = addLongIntegers(lia, lib);

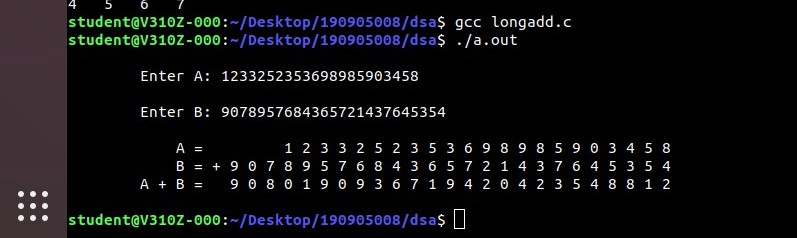
int maxSize = max(lia->value, lib->value, lic->value);

printf("\n\t A = "); displayLongInteger(lia, maxSize); printf("\n\t B = + "); displayLongInteger(lib, maxSize);

printf("\n\tA + B = "); displayLongInteger(lic, maxSize);

printf("\n\n"); return 0;

}



Q2.

#include <stdio.h> #include <stdlib.h> #define MAX 10 typedef struct node {

int key;

struct node \*left, \*right;

}\* NODE;

typedef struct {

NODE S[MAX];

int tos;

}STACK;

NODE newNODE (int item) {

NODE temp = (NODE)malloc(sizeof(struct node)); temp->key = item;

temp->left = temp->right = NULL; return temp;

}

void push (STACK \*s, NODE n) { s->S[++(s->tos)] = n;

}

NODE pop (STACK \*s){

return s->S[(s->tos)--];

}

void inorder (NODE root) { NODE curr;

curr = root; STACK S;

S.tos = -1; push(&S, root); curr = curr->left;

while (S.tos != -1 || curr != NULL) { while (curr != NULL) {

push(&S, curr); curr = curr->left;

}

curr = pop(&S); printf("%d\t", curr->key); curr = curr->right;

}

}

NODE insert (NODE node, int key) {

if (node == NULL) { return newNODE(key);

}

if (key < node->key) {

node->left = insert(node->left, key);

}

else if (key > node->key) {

node->right = insert(node->right, key);

}

return node;

}

NODE minValueNode (NODE node) { NODE current = node;

while (current && current->left != NULL) { current = current->left;

}

return current;

}

NODE deleteNode (NODE root, int key) { if (root == NULL) {

return root;

}

if (key < root->key) {

root->left = deleteNode(root->left, key);

}

else if (key > root->key) {

root->right = deleteNode(root->right, key);

}

else {

if (root->left == NULL) {

NODE temp = root->right; free(root);

return temp;

}

else if (root->right == NULL) { NODE temp = root->left; free(root);

return temp;

}

NODE temp = minValueNode(root->right); root->key = temp->key;

root->right = deleteNode(root->right, temp->key);

}

return root;

}

void main() {

NODE root = NULL;

int k;

printf("Enter the root:\t"); scanf("%d", &k);

root = insert(root, k); int ch;

do {

printf("\nEnter your choice:");

printf("\n1. Create tree\t2. Delete elements\t3. Inorder traversal\t4. Exit:\t");

scanf("%d", &ch); switch (ch) {

case 1: {

printf("Enter element to be inserted:\t"); scanf("%d", &k);

root = insert(root, k); break;

}

case 2: {

printf("Enter element to be deleted:\t"); scanf("%d", &k);

root = deleteNode(root, k); break;

}

case 3: {

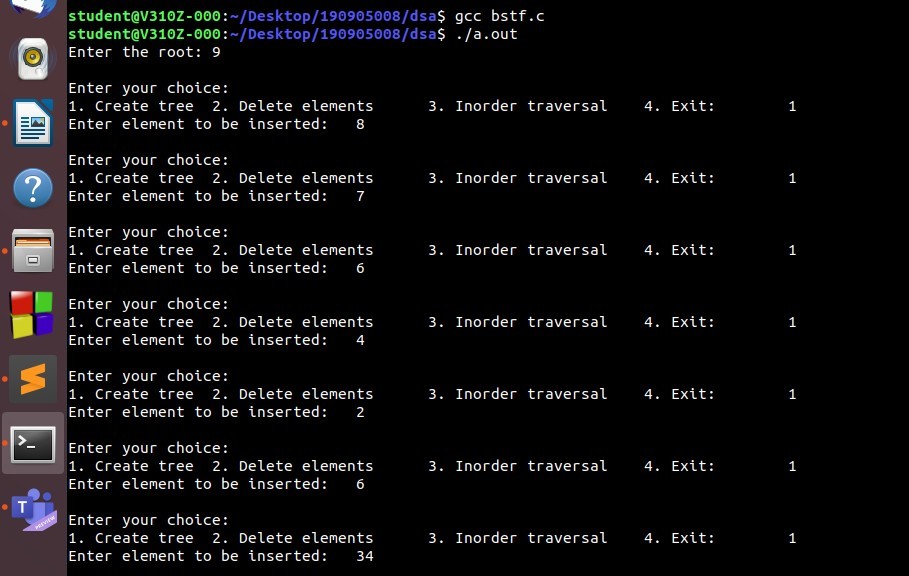
inorder(root); break;

}

}

} while (ch < 4);

}







|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| En be r you r ch one e-  7 . | T nor d er | t r a ver s a T | 4 . | Exi-t 2 |
| En be r eTenenI to be d eTeted- 4 |  |  |  |  |
|  |  |  |  |  |
| En be r you r ch one e-  7 . | T nor d er | t r a ver s a T | 4 . | Exi-t |
|  |  |  |  |  |
| 7 . | T nor d er | t r a ver s a T | 4 . | Exi-t |







01 . c r ea te t r ee 2 . De1 ete e1eme n t s



# 2



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | student@ V z1oz - ooo: | /Desktop/1: |
| En be r you r c h once- |  |  |  |  |  |  |
| 1 . c r ea te t r ee 2 . De1 ete e1eme n t s | 7 . | T n o r d e r | t r a ve r sa T | 4 . | Exi-t |  |
| E n te r e1eme n t t o be I n se r te -fi 89 |  |  |  |  |  |  |
| E n te r y ou r ct o? ce- |  |  |  |  |  |  |
|  | 7 . | T n o r d e r | t r a ve r sa T | 4 . | Exi-t |  |
| E n te r e1eme n t t o be I n se r te -fi 2 ? |  |  |  |  |  |  |
| E n te r y ou r c t o? ce- |  |  |  |  |  |  |
| \* 1 . c r ea te t r ee 2 . De1 ete e1eme n t s | 7 . | T n o r d e r | t r a ve r sa T | 4 . | Exi-t |  |
| E n te r e1eme n t t o be I n se r te -fi é |  |  |  |  |  |  |
| E n te r y ou r ct o? ce- |  |  |  |  |  |  |
| 1 . c r ea te t r ee 2 . De1 ete e1eme n t s  2 7 | 7 . | T n o r d e r | t r a ve r sa T | 4 . | Exi-t |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| E n te r y ou r ct o? c-e  1 . c r ea te t r ee 2 . De1 ete e1eme n t s ’ E n te r e1eme n t t o be fi e1ete -fi 2 | 7 . | T n o r d e r | t r a ve r sa T | 4 . | Exi-t | 2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ” ’ En be r you r ch one e-  I . c r ea te t r ee 2. Delete eT enen t s | 7 . | T nor d er | t r a ver s a T | 4 . | Exi-t | 2 |
| En be r eTenen t to be d eTeted- 2 |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | E n te r y ou r ct o? c-e  1 . c r ea te t r ee 2 . De1 ete e1eme n t s | 7 . | T n o r d e r | t r a ve r sa T | 4 . | Exi-t | 2 |
| En be r eTenen I to be d eTeted- 7 | | | | | | | |
| ’- | E n te r y ou r ct o?c-e  1 . c r ea te t r ee 2 . De1 ete e1eme n t s | 7 . | T n o r d e r | t r a ve r sa T | 4 . | Exi-t | 2 |
| En te r eTenen t to be d eTeted- 2 7 | | | | | | | |
|  | E n te r y ou r ct o? c-e  1 . c r ea te t r ee 2 . De1 ete e1eme n t s | 7 . | T n o r d e r | t r a ve r sa T | 4 . | Exi-t | 2 |
| En te r eTenen t to be d eTeted- 74 | | | | | | | |





7