Final Lab Report -Fall 2022

Course Title: Data Structure Lab

Course Code: SE-132

Submitted To:

Ms. Sazia Sharmin

Lecturer (SWE)

Daffodil International University

Submitted By :

MD : Nurnabi Fahim

ID : 221-35-1049

Section : E

Department : SWE

Daffodil International University

Github Link: **https://github.com/fahim1049/Data-Structure-Lab**

Code 1.1:

<global> | main() : int

Start here | problem 1.c

```c
//code for insert two element at last of the array
#include <stdio.h>
#include <stdlib.h>
int main() {
int *arr, *arr2;
int i, n, m;
printf("Enter the size of primary size of your array: ");
scanf("%d", &n);
arr = (int *)malloc(sizeof(int)*n);
m = n+2;
arr2 = (int *)malloc(sizeof(int)*m);
printf("Enter element of your array:\n");
for(i = 0; i < n; i++) {
scanf("%d", arr+i);
}
printf("Before insert your elements of your array are:\n\t");
for(i = 0; i < n; i++) {
printf("%d ", *(arr+i));
}
for(i = 0 ; i < n ; i++){
*(arr2 + i) = *(arr + i);
}
for(i = n; i < m; i++) {
printf("\nEnter a element for insert last: ");
scanf("%d", arr2+i);
}
printf("\nAfter insert two element at last you array is:\n\t");
for(i = 0; i < m; i++) {
printf("%d ", *(arr2 + i));
}
return 0 ;
}
```

OUTPUT:

```
"C:\Users\Fahim\Desktop\DS report\problem 1.exe"
Enter the size of primary size of your array: 5
Enter element of your array:
2
1
3
4
5
Before insert your elements of your array are:
        2 1 3 4 5
Enter a element for insert last: 6

Enter a element for insert last: 7

After insert two element at last you array is:
        2 1 3 4 5 6 7
Process returned 0 (0x0)   execution time : 47.068 s
Press any key to continue.
```

Discussion: Here, at first this programme will take a number as a initial size of an array. Then it integer value as the element of the array. After that, again it takes 2 more value one by one for add at last of the array. At last it will print the final value.

Code 2.2:

Start here | problem 1.c | p2.c

```c

//insert two differnt numbers at differernt porsition
#include <stdio.h>
int arr[10] = {1, 2, 3, 4, 5, 6, 7};
void insert_element(int pos, int data);
void print();
int main() {
int pos1, pos2, item1, item2;
printf("Before insert:\n");
print();
printf("Enter enter two number for insert in array:\n");
scanf("%d %d", &item1, &item2);
printf("Enter two index:(0-9)\n");
scanf("%d %d", &pos1, &pos2);
if(pos1 > pos2) {
insert_element(pos1, item1);
insert_element(pos2, item2);
}
else {
insert_element(pos2, item2);
insert_element(pos1, item1);
}
printf("After insert data:\n");
print();
return 0;
}
void insert_element(int pos, int data) {
int i;
for(i = 9; i >= pos; i--) {
arr[i] = arr[i-1];
}
arr[i] = data;
}
```

```
void print() {
int i;
for(i = 0; i < 10; i++) {
printf("%d ", arr[i]);
}
printf("\n");
}
```

OUTPUT :

```
"C:\Users\Fahim\Desktop\DS report\p2.exe"
Before insert:
1 2 3 4 5 6 7 0 0 0
Enter enter two number for insert in array:
321
232
Enter two index:(0-9)
4
5
After insert data:
1 2 3 321 4 232 5 6 7 0

Process returned 0 (0x0)   execution time : 55.186 s
Press any key to continue.
```

Discussion:

At first, this program will show you some element of an array which was predefined. Secondly it will asked for two more value and 2 position between zero(0) to nine and last it just add those value in those position and print the array again

Code 3.3:

Start here | problem 1.c | p2.c | p3.c

```c
//applying binary search & delete
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
int main() {
int min = 0, max = 9, mid, del_item, i;
bool flag = true;
int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
printf("Enter which element you want to delete: ");
scanf("%d", &del_item);
while(min <= max) {
mid = (min + max) / 2;
if(del_item == arr[mid]) {
printf("%d found at index %d\n", del_item, mid);
flag = false;
break;
}
else if(arr[mid] < del_item) {
min = mid + 1;
}
else {
max = mid - 1;
}
}
if(flag) {
printf("%d is not available in the data set\n", del_item);
}
else {
for(i = mid; i < 9; i++) {
arr[i] = arr[i+1];
}
```

```
}
printf("After delete: ");
for(i = 0; i < 9; i++) {
printf("%d ", arr[i]);
}
}
return 0;
}
```

Output:

```
"C:\Users\Fahim\Desktop\DS report\p3.exe"
Enter which element you want to delete: 5
5 found at index 4
After delete: 1 2 3 4 6 7 8 9 10
Process returned 0 (0x0)   execution time : 26.080 s
Press any key to continue.
```

Discussion:

At first, this programme will ask you for a integer value to apply binary search. If it is exist in the array user will see a massage " ___ found at index ___" and the data will deleted and print all the data which are exist after deletion. Otherwise it will print another massage "___ is not available in the data set"

Code 4.1 :

Start here | problem 1.c | p2.c | p3.c | p4.c

```c
//comparing linear and binary serch
#include <stdio.h>
#include <stdbool.h>
int main() {
int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int max, min, mid, count_binary = 0, count_linear = 0;
int search_item;
bool flag = true;
printf("Enter which item you want to search(1-10): ");
scanf("%d", &search_item);
printf("\n");
//for bianary
min = 0;
max = 9;
while(min <= max) {
count_binary++;
mid = (min + max) / 2;
if(arr[mid] == search_item) {
printf("%d found after looping %d times\n", search_item,
count_binary);
flag = false;
break;
}
else if(search_item < arr[mid]) {
max = mid - 1;
}
else {
min = mid + 1;
}
}
```

```
for(int i = 0; i < 10; i++) {
count_linear++;
if(arr[i] == search_item) {
printf("%d found atfer looping %d time(s)\n", search_item,
count_linear);
break;
}
}
if(flag) {
printf("Data is not available\n");
}
else {
printf("\n\n====>\t\n\n");
}
return 0;
}
```

Output:

```
"C:\Users\Fahim\Desktop\DS report\p4.exe"
Enter which item you want to search(1-10): 9

9 found after looping 3 times
9 found atfer looping 9 time(s)


.t====>


Process returned 0 (0x0)   execution time : 20.587 s
Press any key to continue.
```

Discussion :

At first, this program want to know about one number 0-10. Then it will start a both search respectively and compare both by counting loop. And so a massage based on the count result.

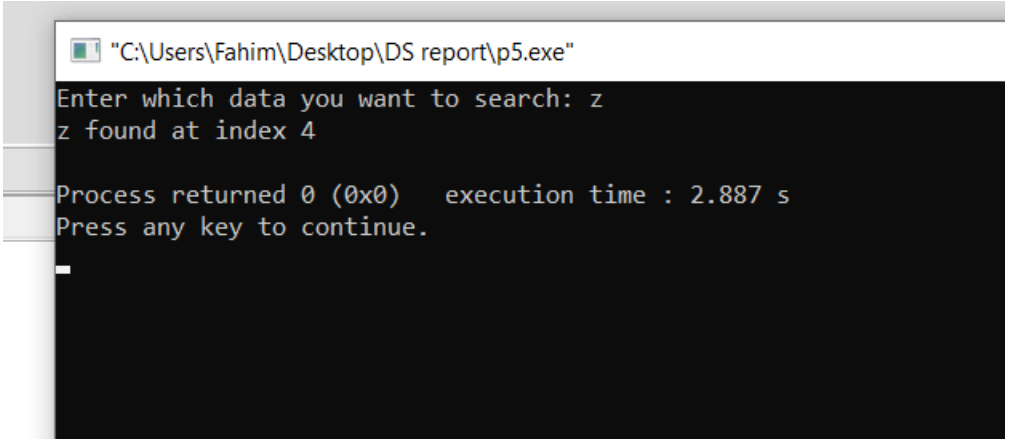
Code 5.1 :

Start here | problem 1.c | p2.c | p3.c | p4.c | p5.c

```c
//Short and binary search
#include <stdio.h>
#include <stdbool.h>
int main() {
char arr[5] = {'z', 'k', 'l', 'a', 'g'};
int i, j, min, max, mid;
char temp, search_data;
bool flag = true;
for(i = 0; i < 5; i++) {
for(j = 0; j < (4 - i); j++) {
if(arr[j] > arr[j+1]) {
temp = arr[j];
arr[j] = arr[j+1];
arr[j+1] = temp;
}
}
}
printf("Enter which data you want to search: ");
fflush(stdin);
scanf("%c", &search_data);
min = 0;
max = 4;
while (min <= max) {
mid = (min + max) / 2;
if(arr[mid] == search_data) {
printf("%c found at index %d\n", search_data, mid);
flag = false;
break;
}
else if(search_data < arr[mid]) {
max = mid - 1;
}
else {
```

```c
min = mid + 1;
}
}
if(flag) {
printf("Data is not in the array\n");
}
return 0;
}
```

Output:



Firstly, this programme will sort it's given data and ask for a character to Search in the array. If match it'll show a massage "--- found at----".

Otherwise, it'll show "Data is not in the array"

Discussion :

Firstly, this programme will sort it's given data and ask for a character to Search in the array. If match it'll show a massage "--- found at----". Otherwise, it'll show "Data is not in the array"

Code 6.1:

Start here | problem 1.c | p2.c | p3.c | p4.c | p5.c | p6.c

```c
//stack
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <stdbool.h>
char arr[5];
int top = -1;
void option();
void insert_stack(char data);
void delete();
void print();
int main() {
int data, choice;
while(true) {
option();
scanf("%d", &choice);
switch(choice) {
case 1:
printf("Enter a character for add in stack: ");
fflush(stdin);
scanf("%c", &data);
insert_stack(data);
break;
case 2:
delete();
break;
case 3:
print();
break;
case 4:
printf("STOPED\n");
exit(0);
default:
```

```
33  default:
34  printf("Invalid input........\n");
35  fflush(stdin);
36  getchar();
37  }
38  }
39  return 0;
40  }
41  void option(){
42  system("cls");
43  printf("1. insert\n");
44  printf("2. delete\n");
45  printf("3. print\n");
46  printf("4. exit\n");
47  printf("\nChoice option: ");
48  }
49  void insert_stack(char data) {
50  top++;
51  if(top >= 5) {
52  printf("=====>Data overflow!!!!!!!!!\n");
53  fflush(stdin);
54  getchar();
55  top = 4;
56  }
57  else {
58  arr[top] = data;
59  }
60  }
61  void delete() {
62  if(top <= -1) {
63  printf("=====>Data underflow!!!!!!!\n");
```

```c
}
}
void delete() {
if(top <= -1) {
printf("=====>Data underflow!!!!!!!\n");
fflush(stdin);
getchar();
}
else {
printf("%c have deleted\n", arr[top]);
fflush(stdin);
getchar();
arr[top] = NULL;
top--;
}
}
void print() {
int i;
if(top == -1) {
printf("Empty stack\n");
}
else {
printf("Data: ");
for(i = top; i >= 0; i--) {
printf("%c", arr[i]);
}
}
fflush(stdin);
getchar();
}
```

OUTPUT:

```
1. insert
2. delete
3. print
4. exit

Choice option: 1
Enter a character for add in stack: E
```

```
1. insert
2. delete
3. print
4. exit

Choice option: 1
Enter a character for add in stack: L
```

```
1. insert
2. delete
3. print
4. exit

Choice option: 1
Enter a character for add in stack: P
```

```
1. insert
2. delete
3. print
4. exit

Choice option: 1
Enter a character for add in stack: P
```

```
1. insert
2. delete
3. print
4. exit

Choice option: 1
Enter a character for add in stack: P
```

Continue……..

Discussion:

This programme is about stack which flow the LIFO method, means Last In First Out. At first, this programme will show us 4 option and ask for choice one of 1. Is for add data in stack 2. is for delete from stack 3. is for print data from stack 4. is for terminate the programme This stack will contain max five element at a time if anybody want to add more than five elements then the programme will show him/her this massage "=====>Data overflow!!!!!!!!!" and also if there will no data in the stack and anyone want to delete from it then he/she will see this massage "=====>Data underflow!!!!!!!!!".

Code 7.1:

```c
//Creating LInked list:
#include <stdio.h>
#include <stdlib.h>
typedef struct node NODE;
struct node {
char ch;
NODE *next;
};
NODE *create_node (char data);
NODE *insert_last(NODE *head, char data);
void print(NODE *head);
int main() {
char item;
int choice = 1;
NODE *head = NULL;
while (choice) {
printf("Enter a character to add in linked list: ");
fflush(stdin);
scanf("%c", &item);
head = insert_last(head, item);
printf("Do you want to continue \n(press zero(0) for exit:");
scanf("%d", &choice);
}
printf("Your linked list is: ");
print(head);
printf("\nEnter new character to insert at last of your list: ");
fflush(stdin);
scanf("%c", &item);
head = insert_last(head, item);
printf("\nAfter isertion your list is: ");
print(head);
return 0;
}
```

```c
}
NODE *create_node(char data) {
NODE *new_node = (NODE*)malloc(sizeof(NODE));
if(new_node != NULL) {
new_node->ch = data;
new_node->next = NULL;
}
return new_node;
}
NODE *insert_last(NODE *head, char item) {
NODE *current_node = create_node(item);
if(head == NULL) {
return current_node;
}
NODE *temp = head;
while(temp->next != NULL) {
temp = temp->next;
}
temp->next = current_node;
return head;
}
void print(NODE *head) {
if(head == NULL) {
printf("List is empty!\n");
}
else {
NODE *temp = head;
while(temp != NULL) {
printf("[%x]===>[%c]", temp, temp->ch);
temp = temp->next;
}
```

```c
temp->next = current_node;
return head;
}
void print(NODE *head) {
if(head == NULL) {
printf("List is empty!\n");
}
else {
NODE *temp = head;
while(temp != NULL) {
printf("[%x]===>[%c]", temp, temp->ch);
temp = temp->next;
}
printf("[NULL]\n");
}
}
```

OUTPUT:

```
"C:\Users\Fahim\Desktop\DS report\p7.exe"
Enter a character to add in linked list: A
Do you want to continue
(press zero(0) for exit:1
Enter a character to add in linked list: B
Do you want to continue
(press zero(0) for exit:1
Enter a character to add in linked list: C
Do you want to continue
(press zero(0) for exit:0
Your linked list is: [b26a10]===>[A][b26a30]===>[B][b26a50]===>[C][NULL]

Enter new character to insert at last of your list: D

After isertion your list is: [b26a10]===>[A][b26a30]===>[B][b26a50]===>[C][b26a70]===>[D][NULL]

Process returned 0 (0x0)   execution time : 19.632 s
Press any key to continue.
```

Discussion:

At first, this programme will ask for enter a charackter to insert in linked list. And ask user is he/she want to continue? If he/she want to continue again it will ask for a character and it will continue while he/she press zero (0). If press 0 then it will print all the value he/she input and it’s address in hexadecimal. After that it will be wait for again give a data for insert a last and again it print life previous with new data. Finally, it will terminate.

Code8.1:

Start here | problem 1.c | p2.c | p3.c | p4.c | p5.c | p6.c | p7.c | p8.c

```c
//implementation of queue
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
typedef struct node Q_NODE;
struct node {
int data;
Q_NODE *next;
};
typedef struct counter_node C_NODE;
struct counter_node {
int tail;
};
Q_NODE *create_node(int data);
Q_NODE *add_node_last(Q_NODE *head , int data);
Q_NODE *delete_first(Q_NODE *head);
void print(Q_NODE *head);
void option();
int main() {
Q_NODE *head = NULL;
C_NODE count;
count.tail = 0;
int q_size, optn, data;
printf("Enter the size of your Queue: ");
scanf("%d", &q_size);
while(true){
option();
printf("\n\nchoice your option: ");
scanf("%d", &optn);
switch(optn) {
```

```c
case 1:
{
printf("Enter a data for insert: ");
scanf("%d", &data);
count.tail++;
if(count.tail > q_size) {
printf("Queue is full. Data overFlow\n");
count.tail--;
printf("\npress any key to Countinue\n\n");
fflush(stdin);
getchar();
}
else {
head = add_node_last(head, data);
}
}
break;
case 2:
{
count.tail--;
if(count.tail < 0) {
count.tail++;
}
```

```
53  if(count.tail < 0) {
54  count.tail++;
55  }
56  else {
57  head = delete_first(head);
58  }
59  }
60  printf("\n\npress any key to continue.........");
61  fflush(stdin);
62  getchar();
63  break;
64  case 3:
65  printf("You have: ");
66  print(head);
67  break;
68  case 4:
69  exit(0);
70  default:
71  printf("Invalid input..........\n\n\n");
72  }
73  }
74  return 0;
75  }
76  void option() {
77  printf("1. Insert\n");
78  printf("2. Delete\n");
79  printf("3. Print queue\n");
80  printf("4. Exit");
81  }
82  Q_NODE *create_node(int data) {
```

```c
}
Q_NODE *create_node(int data) {
Q_NODE *new_node = (Q_NODE*)malloc(sizeof(Q_NODE));
if(new_node != NULL) {
new_node->data = data;
new_node->next = NULL;
}
return new_node;
}
Q_NODE *add_node_last(Q_NODE *head, int data) {
Q_NODE *current_node = create_node(data);
if(head == NULL) {
return current_node;
}
Q_NODE *temp = head;
while(temp->next != NULL) {
temp = temp->next;
}
temp->next = current_node;
return head;
}
Q_NODE *delete_first(Q_NODE *head) {
if(head == NULL) {
printf("\nData underfloaw");
}
else {
printf("%d is deleted\n", head->data);
head = head->next;
}
return head;
}
void print(Q_NODE *head) {
```

```c
if(head == NULL) {
printf("Empty Queue\n");
}
else {
Q_NODE *temp = head;
while(temp != NULL) {
printf("%d ", temp->data);
temp = temp->next;
}
}
printf("\n");
printf("press any key for countinue.......\n\n");
fflush(stdin);
getchar();
}
```

Output:

```
"C:\Users\Fahim\Desktop\DS report\p8.exe"
Enter the size of your Queue: 3
1. Insert
2. Delete
3. Print queue
4. Exit

choice your option: 1
Enter a data for insert: 200
1. Insert
2. Delete
3. Print queue
4. Exit

choice your option: 3
You have: 200
press any key for countinue.......
```

Discussion:

This programme is follow queue structure which is follow FIFO, means First In First Out method. At first, this programme will show a massage for take the size of the Queue. Then it will show us 4 option and ask for choice one of 1. Is for add data in stack 2. is for delete from stack 3. is for print data from stack 4. is for terminate the programme Similarly, like stack it is option are work and this program also handle the overflow and underflow.

Code 9.1:

Start here | problem 1.c | *p2.c | p3.c | p4.c | p5.c | p6.c | p7.c | p8.c | p9.c

```c

//binary search tree
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
typedef struct node tree_node;
struct node {
char value;
tree_node *left;
tree_node *right;
};
//all fucntion decleare here
tree_node *create_node(char item);
tree_node *add_node(tree_node *root, char item);
void print_pre(tree_node *root);
void print_in(tree_node *root);
void print_post(tree_node *root);
//main function
int main() {
tree_node *root = NULL;
int choice;
char item;
while(true) {
printf("Enter a data to add in tree: ");
fflush(stdin);
scanf("%c", &item);
root = add_node(root, item);
printf("Do you want to continue: ");
scanf("%d", &choice);
if(!choice) {
break;
```

```
}
}
printf("Pre order: ");
print_pre(root);
printf("\nInorder: ");
print_in(root);
printf("\nPost Order: ");
print_post(root);
return 0;
}
//node create function
tree_node *create_node(char item) {
tree_node *new_node = (tree_node*)malloc(sizeof(tree_node));
if(new_node != NULL) {
new_node->value = item;
new_node->left = NULL;
new_node->right = NULL;
}
return new_node;
}
//node add function
tree_node *add_node(tree_node *root, char item){
tree_node *temp = root;
tree_node *current_node = create_node(item);
if(temp == NULL) {
return current_node;
}
if(item < temp->value) {
temp->left = add_node(temp->left, item);
}
else {
temp->right = add_node(temp->right, item);
```

```c
}
return temp;
}
//print pre order function
void print_pre(tree_node *root) {
tree_node *temp = root;
if(temp == NULL) {
return;
}
printf("%c ", temp->value);
print_pre(temp->left);
print_pre(temp->right);
}
//print in order function
void print_in(tree_node *root) {
tree_node *temp = root;
if(temp == NULL) {
return;
}
print_in(temp->left);
printf("%c ", temp->value);
print_in(temp->right);
}
//print post order fuction
void print_post(tree_node *root) {
tree_node *temp = root;
if(temp == NULL) {
return;
}
print_post(temp->left);
```

```c
}
print_in(temp->left);
printf("%c ", temp->value);
print_in(temp->right);
}
//print post order fuction
void print_post(tree_node *root) {
tree_node *temp = root;
if(temp == NULL) {
return;
}
print_post(temp->left);
print_post(temp->right);
printf("%c ", temp->value);
}
```

Output:

```
Enter a data to add in tree: 50
Do you want to continue: 1
Enter a data to add in tree: 40
Do you want to continue: 1
Enter a data to add in tree: 70
Do you want to continue: 1
Enter a data to add in tree: 35
Do you want to continue: 1
Enter a data to add in tree: 45
Do you want to continue: 1
Enter a data to add in tree: 60
Do you want to continue: 1
Enter a data to add in tree: 75
Do you want to continue: 0
Pre order: 50  40  35  45  70  60  75
Inorder: 35  40  45  50  60  70  75
Post Order: 35  45  40  60  75  70  50
```

Discussion:

This programme is as similar as 9.1. The difference between this and previous programme is it is work with integer and previously one is with character. That’s is