#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Structure to represent a day

struct Day {

char \*name;

int date;

char \*activity;

};

// Function to create a day

struct Day createDay() {

struct Day day;

// Allocate memory for day name

day.name = (char \*)malloc(sizeof(char) \* 20);

printf("Enter day name: ");

scanf("%s", day.name);

printf("Enter date: ");

scanf("%d", &day.date);

// Allocate memory for activity description

day.activity = (char \*)malloc(sizeof(char) \* 100);

printf("Enter activity: ");

scanf(" %[^\n]s", day.activity);

return day;

}

// Function to read the calendar data from the keyboard

void readCalendar(struct Day \*calendar, int size) {

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

calendar[i] = createDay();

}

}

// Function to display the calendar

void displayCalendar(struct Day \*calendar, int size) {

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

printf("Day: %s\n", calendar[i].name);

printf("Date: %d\n", calendar[i].date);

printf("Activity: %s\n\n", calendar[i].activity);

}

}

// Function to free memory allocated for the calendar

void freeCalendar(struct Day \*calendar, int size) {

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

free(calendar[i].name);

free(calendar[i].activity);

}

}

int main() {

int size;

// Get the number of days in a week

printf("Enter the number of days in a week: ");

scanf("%d", &size);

// Dynamically allocate memory for the calendar

struct Day \*calendar = (struct Day \*)malloc(sizeof(struct Day) \* size);

// Check if memory allocation is successful

if (calendar == NULL) {

printf("Memory allocation failed.\n");

return 1; // Exit the program with an error code

}

// Read calendar data from the keyboard

readCalendar(calendar, size);

// Display calendar details

printf("\nCalendar Details:\n");

displayCalendar(calendar, size);

// Free memory allocated for the calendar

freeCalendar(calendar, size);

free(calendar);

return 0; // Exit the program successfully

}

Program 2: Develop a Program in C for the following operations on strings:

a.Read a main string(STR), a Pattern String(PAT) and Replace String (REP) .

b- Perform Pattern matching Operation: find and replace all occurances of PAT does not exist in STR.

support the program with functions for each of the above operations. Dont use built in functions.

#include <stdio.h>

#include <string.h>

// Function to read a string

void readString(char \*str, const char \*prompt) {

printf("%s", prompt);

scanf("%s", str);

}

// Function to perform pattern matching and replacement

void patternMatching(char \*mainStr, const char \*pattern, const char \*replace) {

char result[1000]; // Assuming a fixed size for the result, adjust as needed

int mainLen = strlen(mainStr);

int patLen = strlen(pattern);

int repLen = strlen(replace);

int i, j, k;

for (i = 0; i <= mainLen - patLen; ) {

j = 0;

// Check for pattern match

while (j < patLen && mainStr[i + j] == pattern[j]) {

j++;

}

// If pattern found, replace it and move index accordingly

if (j == patLen) {

for (k = 0; k < repLen; k++) {

result[i + k] = replace[k];

}

i += repLen;

} else {

result[i] = mainStr[i];

i++;

}

}

// Copy remaining characters from mainStr to result

while (i < mainLen) {

result[i] = mainStr[i];

i++;

}

// Null-terminate the result

result[i] = '\0';

// Copy result back to mainStr

strcpy(mainStr, result);

}

int main() {

char mainStr[1000], pattern[100], replace[100];

// Read main string

readString(mainStr, "Enter the main string: ");

// Read pattern string

readString(pattern, "Enter the pattern string: ");

// Read replace string

readString(replace, "Enter the replace string: ");

// Perform pattern matching and replacement

patternMatching(mainStr, pattern, replace);

// Display the result

printf("Result after pattern matching and replacement: %s\n", mainStr);

return 0;

}

3. Develop a menu driven program in C for the following operations on STACK of integers (Array Implementation of stack with Maximum size MAX)

a. Push an element on to stack

b. Pop an element from stack

c. Demonstrate how stack can be used to check Palindrome

d.Demonstrate Overflow and Underflow situations on stack

e.Display the status of stack

f. exit.

Support the program with appropriate functions for each of the above operation

#include<stdio.h>

#include<stdlib.h>

#define MAX 5

int s[MAX];

int top = -1;

void push(int item);

int pop();

void palindrome();

void display();

void main()

{

            int choice, item;

            while(1)

            {

                        printf("\n\n\n\n~~~~~~Menu~~~~~~ : ");

                        printf("\n=>1.Push an Element to Stack and Overflow demo ");

                        printf("\n=>2.Pop an Element from Stack and Underflow demo");

                        printf("\n=>3.Palindrome demo ");

                        printf("\n=>4.Display ");

                        printf("\n=>5.Exit");

                        printf("\nEnter your choice: ");

                        scanf("%d", &choice);

                        switch(choice)

                        {

                                    case 1:             printf("\nEnter an element to be pushed: ");

                                                            scanf("%d", &item);

**push(item);**

                                                            break;

                                    case 2:             **item = pop();**

                                                            if(item != -1)

                                                                        printf("\nElement popped is: %d", item);

                                                            break;

                                    case 3:             **palindrome();**

                                                            break;

                                    case 4:             **display();**

                                                            break;

                                    case 5:             exit(1);

                                    default:            printf("\nPlease enter valid choice ") ;

                                                            break;

                    }

            }

}

void push(int item)

{

            if(top == MAX-1)

            {

                        printf("\n~~~~Stack overflow~~~~");

                        return;

            }

            top = top + 1 ;

            s[top] = item;

}

int pop()

{

            int item;

            if(top == -1)

            {

                        printf("\n~~~~Stack underflow~~~~");

                        return -1;

            }

            item = s[top];

            top = top - 1;

            return item;

}

void display()

{

            int i;

            if(top == -1)

            {

                        printf("\n~~~~Stack is empty~~~~");

                        return;

            }

            printf("\nStack elements are:\n ");

            for(i=top; i>=0 ; i--)

                        printf("| %d |\n", s[i]);

}

void palindrome()

{

            int flag=1,i;

            printf("\nStack content are:\n");

            for(i=top; i>=0 ; i--)

                        printf("| %d |\n", s[i]);

            printf("\nReverse of stack content are:\n");

            for(i=0; i<=top; i++)

                        printf("| %d |\n", s[i]);

            for(i=0; i<=top/2; i++)

            {

                        if( s[i] != s[top-i] )

                        {

                                    flag = 0;

                                    break;

                        }

            }

            if(flag == 1)

            {

                        printf("\nIt is palindrome number");

            }

            else

            {

                        printf("\nIt is not a palindrome number");

            }

}

3. Develop a menu driven program in C for the following operations on STACK of integers (Array Implementation of stack with Maximum size MAX)

a. Push an element on to stack

b. Pop an element from stack

c. Demonstrate how stack can be used to check Palindrome

d.Demonstrate Overflow and Underflow situations on stack

e.Display the status of stack

f. exit.

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#define MAX\_SIZE 100

// Structure to represent the stack

struct Stack {

int items[MAX\_SIZE];

int top;

};

// Function prototypes

void initialize(struct Stack \*s);

void push(struct Stack \*s, int value);

int pop(struct Stack \*s);

bool isPalindrome(struct Stack \*s);

void display(struct Stack \*s);

bool isFull(struct Stack \*s);

bool isEmpty(struct Stack \*s);

int main() {

struct Stack stack;

initialize(&stack);

int choice, value;

do {

// Display menu

printf("\nMenu:\n");

printf("1. Push\n");

printf("2. Pop\n");

printf("3. Check Palindrome\n");

printf("4. Display Stack\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

if (isFull(&stack)) {

printf("Stack Overflow! Cannot push more elements.\n");

} else {

printf("Enter the value to push: ");

scanf("%d", &value);

push(&stack, value);

printf("Element %d pushed onto the stack.\n", value);

}

break;

case 2:

if (isEmpty(&stack)) {

printf("Stack Underflow! Cannot pop from an empty stack.\n");

} else {

value = pop(&stack);

printf("Popped element: %d\n", value);

}

break;

case 3:

if (isPalindrome(&stack)) {

printf("The stack elements form a palindrome.\n");

} else {

printf("The stack elements do not form a palindrome.\n");

}

break;

case 4:

display(&stack);

break;

case 5:

printf("Exiting the program.\n");

break;

default:

printf("Invalid choice. Please enter a valid option.\n");

}

} while (choice != 5);

return 0;

}

// Function to initialize the stack

void initialize(struct Stack \*s) {

s->top = -1;

}

// Function to push an element onto the stack

void push(struct Stack \*s, int value) {

s->items[++s->top] = value;

}

// Function to pop an element from the stack

int pop(struct Stack \*s) {

return s->items[s->top--];

}

// Function to check if the stack elements form a palindrome

bool isPalindrome(struct Stack \*s) {

int i, j;

for (i = 0, j = s->top; i <= j; i++, j--) {

if (s->items[i] != s->items[j]) {

return false;

}

}

return true;

}

// Function to display the elements of the stack

void display(struct Stack \*s) {

if (isEmpty(s)) {

printf("Stack is empty.\n");

} else {

printf("Stack elements: ");

for (int i = 0; i <= s->top; i++) {

printf("%d ", s->items[i]);

}

printf("\n");

}

}

// Function to check if the stack is full

bool isFull(struct Stack \*s) {

return s->top == MAX\_SIZE - 1;

}

// Function to check if the stack is empty

bool isEmpty(struct Stack \*s) {

return s->top == -1;

}

Program 4: Develop a program in C for converting an Infix expression to Postfix expression. Program shouls support for both parenthesezed expressions with the operators +,-,\*,/, ^ and alphanumeric operands.

#include<stdio.h>

#include<ctype.h>

char stack[100];

int top = -1;

void push(char x)

{

stack[++top] = x;

}

char pop()

{

if(top == -1)

return -1;

else

return stack[top--];

}

int priority(char x)

{

if(x == '(')

return 0;

if(x == '+' || x == '-')

return 1;

if(x == '\*' || x == '/')

return 2;

return 0;

}

int main()

{

char exp[100];

char \*e, x;

printf("Enter the expression : ");

scanf("%s",exp);

printf("\n");

e = exp;

while(\*e != '\0')

{

if(isalnum(\*e))

printf("%c ",\*e);

else if(\*e == '(')

push(\*e);

else if(\*e == ')')

{

while((x = pop()) != '(')

printf("%c ", x);

}

else

{

while(priority(stack[top]) >= priority(\*e))

printf("%c ",pop());

push(\*e);

}

e++;

}

while(top != -1)

{

printf("%c ",pop());

}return 0;

}

1. Develop a c program for the following stack applications a) evaluation of suffix expression with single digit operands and operators +,-,\*,/,%,^

b)solving tower honoi problem with n disks

#include<stdio.h>

#include<math.h>

#include<string.h>

#include<ctype.h>

double compute(char symbol,double op1,double op2);

void main()

{

double s[20],res,op1,op2;

int top=-1,i;

char postfix[20],symbol;

printf("Enter the Postfix Expression\n");

scanf("%s",postfix);

for(i=0;i<strlen(postfix);i++)

{

symbol=postfix[i];

if(isdigit(symbol))

s[++top]=symbol-'0';

else

{

op2=s[top--];

op1=s[top--];

res=compute(symbol,op1,op2);

s[++top]=res;

}

}

res=s[top--];

printf("The result %f \n",res);

}

double compute(char symbol,double op1,double op2)

{

switch(symbol)

{

case '+':return op1+op2;

case '-':return op1-op2;

case '/':return op1/op2;

case '\*':return op1\*op2;

/\*case '^':

case '$':return (pow(op1,op2)) ;\*/

}

}

B .

#include<stdio.h>

#include<stdlib.h>

void TOH( int n, char A, char B, char C);

void main()

{

int n;

printf("Enter number of rings:\n");

scanf("%d", &n);

TOH(n, 'A', 'C', 'B');

}

void TOH( int n, char A, char B, char C)

{

if(n==1)

printf("Move from %c to %c\n",A, B);

else

{

TOH(n-1, A, C, B);

TOH(1, A, B, C);

TOH(n-1, C, B, A);

}

}

Output

Enter number of rings:

3

Move from A to C

Move from A to B

Move from C to B

Move from A to C

Move from B to A

Move from B to C

Move from A to C

6. Develop a menu driven Program in C for the following operations on Circular QUEUE of

Characters (Array Implementation of Queue with maximum size MAX)

a. Insert an Element on to Circular QUEUE

b. Delete an Element from Circular QUEUE

c. Demonstrate Overflow and Underflow situations on Circular QUEUE

d. Display the status of Circular QUEUE

e. Exit

Support the program with appropriate functions for each of the above operations

#include <stdio.h>

#include <stdlib.h>

#define max 5

int q[max],f=-1,r=-1;

void ins()

{

if(f==(r+1)%max)

printf("\nQueue overflow");

else

{

if(f==-1)

f++;

r=(r+1)%max;

printf("\nEnter element to be inserted:");

scanf("%d",&q[r]);

}

}

void del()

{

if(r==-1)

printf("\nQueue underflow");

else

{

printf("\nElemnt deleted is:%d",q[f]);

if(f==r)

f=r=-1;

else

f=(f+1)%max;

}

}

void disp()

{

if(f==-1)

printf("\nQueue empty");

else

{

int i;

printf("\nQueue elements are:\n");

for(i=f;i!=r;i=(i+1)%max)

printf("%d\t",q[i]);

printf("%d",q[i]);

printf("\nFront is at:%d\nRear is at:%d",q[f],q[r]);

}

}

int main()

{

printf("\nCircular Queue operations");

printf("\n1.Insert");

printf("\n2.Delete");

printf("\n3.Display");

printf("\n4.Exit");

int ch;

do{

printf("\nEnter choice:");

scanf("%d",&ch);

switch(ch)

{

case 1:ins();break;

case 2:del();break;

case 3:disp();break;

case 4:exit(0);

default:printf("\nInvalid choice...!");

}

}while(1);

return 0;

}

Lab Program 7:

Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Branch, Sem, PhNo

a. Create a SLL of N Students Data by using front insertion.

b. Display the status of SLL and count the number of nodes in it

c. Perform Insertion / Deletion at End of SLL

d. Perform Insertion / Deletion at Front of SLL(Demonstration of stack)

e. Exit

#include<stdio.h>

#include<stdlib.h>

struct node

{

char usn[25],name[25],branch[25];

int sem;

long int phone;

struct node \*link;

};

typedef struct node \* NODE;

NODE start = NULL;

int count=0;

NODE create()

{

NODE snode;

snode = (NODE)malloc(sizeof(struct node));

if(snode == NULL)

{

printf("\nMemory is not available");

exit(1);

}

printf("\nEnter the usn,Name,Branch, sem,PhoneNo of the student:");

scanf("%s %s %s %d %ld",snode->usn, snode->name, snode->branch, &snode->sem, &snode->phone);

snode->link=NULL;

count++;

return snode;

}

NODE insertfront()

{

NODE temp;

temp = create();

if(start == NULL)

{

return temp;

}

temp->link = start;

return temp;

}

NODE deletefront()

{

NODE temp;

if(start == NULL)

{

printf("\nLinked list is empty");

return NULL;

}

if(start->link == NULL)

{

printf("\nThe Student node with usn:%s is deleted ",start->usn);

count--;

free(start);

return NULL;

}

temp = start;

start = start->link;

printf("\nThe Student node with usn:%s is deleted",temp->usn);

count--;

free(temp);

return start;

}

NODE insertend()

{

NODE cur,temp;

temp = create();

if(start == NULL)

{

return temp;

}

cur = start;

while(cur->link !=NULL)

{

cur = cur->link;

}

cur->link = temp;

return start;

}

NODE deleteend()

{

NODE cur,prev;

if(start == NULL)

{

printf("\nLinked List is empty");

return NULL;

}

if(start->link == NULL)

{

printf("\nThe student node with the usn:%s is deleted",start->usn);

free(start);

count--;

return NULL;

}

prev = NULL;

cur = start;

while(cur->link!=NULL)

{

prev = cur;

cur = cur->link;

}

printf("\nThe student node with the usn:%s is deleted",cur->usn);

free(cur);

prev->link = NULL;

count--;

return start;

}

void display()

{

NODE cur;

int num=1;

if(start == NULL)

{

printf("\nNo Contents to display in SLL \n");

return;

}

printf("\nThe contents of SLL: \n");

cur = start;

while(cur!=NULL)

{

printf("\n||%d|| USN:%s| Name:%s| Branch:%s| Sem:%d| Ph:%ld|",num,cur->usn, cur->name,cur->branch, cur->sem,cur->phone);

cur = cur->link;

num++;

}

printf("\n No of student nodes is %d \n",count);

}

void stackdemo()

{

int ch;

while(1)

{

printf("\n~~~Stack Demo using SLL~~~\n");

printf("\n1:Push operation \n2: Pop operation \n3: Display \n4:Exit \n");

printf("\nEnter your choice for stack demo");

scanf("%d",&ch);

switch(ch)

{

case 1: start = insertfront();

break;

case 2: start = deletefront();

break;

case 3: display();

break;

default : return;

}

}

return;

}

int main()

{

int ch,i,n;

while(1)

{

printf("\n~~~Menu~~~");

printf("\nEnter your choice for SLL operation \n");

printf("\n1:Create SLL of Student Nodes");

printf("\n2:DisplayStatus");

printf("\n3:InsertAtEnd");

printf("\n4:DeleteAtEnd");

printf("\n5:Stack Demo using SLL(Insertion and Deletion at Front)");

printf("\n6:Exit \n");

printf("\nEnter your choice:");

scanf("%d",&ch);

switch(ch)

{

case 1 : printf("\nEnter the no of students: ");

scanf("%d",&n);

for(i=1;i<=n;i++)

start = insertfront();

break;

case 2: display();

break;

case 3: start = insertend();

break;

case 4: start = deleteend();

break;

case 5: stackdemo();

break;

case 6: exit(0);

default: printf("\nPlease enter the valid choice");

}

}

}

Output:

~~~Menu~~~

Enter your choice for SLL operation

1:Create SLL of Student Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:Stack Demo using SLL(Insertion and Deletion at Front)

6:Exit

Enter your choice:1

Enter the no of students: 3

Enter the usn,Name,Branch, sem,PhoneNo of the student:

111

aaa

cs

1

111111

Enter the usn,Name,Branch, sem,PhoneNo of the student:

222

bbb

ec

2

222222

Enter the usn,Name,Branch, sem,PhoneNo of the student:

333

ccc

ec

3

333333

~~~Menu~~~

Enter your choice for SLL operation

1:Create SLL of Student Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:Stack Demo using SLL(Insertion and Deletion at Front)

6:Exit

Enter your choice:2

The contents of SLL:

||1|| USN:333| Name:ccc| Branch:ec| Sem:3| Ph:333333|

||2|| USN:222| Name:bbb| Branch:ec| Sem:2| Ph:222222|

||3|| USN:111| Name:aaa| Branch:cs| Sem:1| Ph:111111|

No of student nodes is 3

~~~Menu~~~

Enter your choice for SLL operation

1:Create SLL of Student Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:Stack Demo using SLL(Insertion and Deletion at Front)

6:Exit

Enter your choice:3

Enter the usn,Name,Branch, sem,PhoneNo of the student:

444

ddd

ec

4

444444

~~~Menu~~~

Enter your choice for SLL operation

1:Create SLL of Student Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:Stack Demo using SLL(Insertion and Deletion at Front)

6:Exit

Enter your choice:2

The contents of SLL:

||1|| USN:333| Name:ccc| Branch:ec| Sem:3| Ph:333333|

||2|| USN:222| Name:bbb| Branch:ec| Sem:2| Ph:222222|

||3|| USN:111| Name:aaa| Branch:cs| Sem:1| Ph:111111|

||4|| USN:444| Name:ddd| Branch:ec| Sem:4| Ph:444444|

No of student nodes is 4

~~~Menu~~~

Enter your choice for SLL operation

1:Create SLL of Student Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:Stack Demo using SLL(Insertion and Deletion at Front)

6:Exit

Enter your choice:4

The student node with the usn: 444 is deleted

~~~Menu~~~

Enter your choice for SLL operation

1:Create SLL of Student Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:Stack Demo using SLL(Insertion and Deletion at Front)

6:Exit

Enter your choice:2

The contents of SLL:

||1|| USN:333| Name:ccc| Branch:ec| Sem:3| Ph:333333|

||2|| USN:222| Name:bbb| Branch:ec| Sem:2| Ph:222222|

||3|| USN:111| Name:aaa| Branch:cs| Sem:1| Ph:111111|

No of student nodes is 3

~~~Menu~~~

Enter your choice for SLL operation

1:Create SLL of Student Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:Stack Demo using SLL(Insertion and Deletion at Front)

6:Exit

Enter your choice:4

The student node with the usn: 111 is deleted

~~~Menu~~~

Enter your choice for SLL operation

1:Create SLL of Student Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:Stack Demo using SLL(Insertion and Deletion at Front)

6:Exit

Enter your choice:5

~~~Stack Demo using SLL~~~

1:Push operation

2: Pop operation

3: Display

4:Exit

Enter your choice for stack demo: 1

Enter the usn,Name,Branch, sem,PhoneNo of the student:

555

eee

cs

1

555555

~~~Stack Demo using SLL~~~

1:Push operation

2: Pop operation

3: Display

4:Exit

Enter your choice for stack demo:3

The contents of SLL:

||1|| USN:555| Name:eee| Branch:cs| Sem:1| Ph:555555|

||2|| USN:333| Name:ccc| Branch:ec| Sem:3| Ph:333333|

||3|| USN:222| Name:bbb| Branch:ec| Sem:2| Ph:222222|

No of student nodes is 3

~~~Stack Demo using SLL~~~

1:Push operation

2: Pop operation

3: Display

4:Exit

Enter your choice for stack demo: 1

Enter the usn,Name,Branch, sem,PhoneNo of the student:

666

fff

cs

6

666666

~~~Stack Demo using SLL~~~

1:Push operation

2: Pop operation

3: Display

4:Exit

Enter your choice for stack demo: 3

The contents of SLL:

||1|| USN:666| Name:fff| Branch:cs| Sem:6| Ph:666666|

||2|| USN:555| Name:eee| Branch:cs| Sem:1| Ph:555555|

||3|| USN:333| Name:ccc| Branch:ec| Sem:3| Ph:333333|

||4|| USN:222| Name:bbb| Branch:ec| Sem:2| Ph:222222|

No of student nodes is 4

~~~Stack Demo using SLL~~~

1:Push operation

2: Pop operation

3: Display

4:Exit

Enter your choice for stack demo: 2

The Student node with usn: 666 is deleted

~~~Stack Demo using SLL~~~

1:Push operation

2: Pop operation

3: Display

4:Exit

Enter your choice for stack demo: 3

The contents of SLL:

||1|| USN:555| Name:eee| Branch:cs| Sem:1| Ph:555555|

||2|| USN:333| Name:ccc| Branch:ec| Sem:3| Ph:333333|

||3|| USN:222| Name:bbb| Branch:ec| Sem:2| Ph:222222|

No of student nodes is 3

~~~Stack Demo using SLL~~~

1:Push operation

2: Pop operation

3: Display

4:Exit

Enter your choice for stack demo: 4

~~~Menu~~~

Enter your choice for SLL operation

1:Create SLL of Student Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:Stack Demo using SLL(Insertion and Deletion at Front)

6:Exit

Enter your choice:6

Lab program 8:

Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo

a. Create a DLL of N Employees Data by using end insertion.

b. Display the status of DLL and count the number of nodes in it

c. Perform Insertion and Deletion at End of DLL

d. Perform Insertion and Deletion at Front of DLL

e. Demonstrate how this DLL can be used as Double Ended Queue

f. Exit

#include<stdio.h>

#include<stdlib.h>

struct node

{

char ssn[25],name[25],dept[10],designation[25];

int sal;

long int phone;

struct node \*llink;

struct node \*rlink;

};

typedef struct node\* NODE;

NODE first = NULL;

int count=0;

NODE create()

{

NODE enode;

enode = (NODE)malloc(sizeof(struct node));

if( enode== NULL)

{

printf("\nRunning out of memory");

exit(0);

}

printf("\nEnter the ssn,Name,Department,Designation,Salary,PhoneNo of the employee: \n");

scanf("%s %s %s %s %d %ld", enode->ssn, enode->name, enode->dept, enode->designation, &enode->sal, &enode->phone);

enode->llink=NULL;

enode->rlink=NULL;

count++;

return enode;

}

NODE insertfront()

{

NODE temp;

temp = create();

if(first == NULL)

{

return temp;

}

temp->rlink = first;

first->llink = temp;

return temp;

}

void display()

{

NODE cur;

int nodeno=1;

cur = first;

if(cur == NULL)

printf("\nNo Contents to display in DLL");

while(cur!=NULL)

{

printf("\nENode:%d||SSN:%s|Name:%s|Department:%s|Designation:%s|Salary:%d|Phone no:%ld", nodeno, cur->ssn, cur->name,cur->dept, cur->designation, cur->sal, cur->phone);

cur = cur->rlink;

nodeno++;

}

printf("\nNo of employee nodes is %d",count);

}

NODE deletefront()

{

NODE temp;

if(first == NULL)

{

printf("\nDoubly Linked List is empty");

return NULL;

}

if(first->rlink== NULL)

{

printf("\nThe employee node with the ssn:%s is deleted", first->ssn);

free(first);

count--;

return NULL;

}

temp = first;

first = first->rlink;

temp->rlink = NULL;

first->llink = NULL;

printf("\nThe employee node with the ssn:%s is deleted",temp->ssn);

free(temp);

count--;

return first;

}

NODE insertend()

{

NODE cur, temp;

temp = create();

if(first == NULL)

{

return temp;

}

cur= first;

while(cur->rlink!=NULL)

{

cur = cur->rlink;

}

cur->rlink = temp;

temp->llink = cur;

return first;

}

NODE deleteend()

{

NODE prev,cur;

if(first == NULL)

{

printf("\nDoubly Linked List is empty");

return NULL;

}

if(first->rlink == NULL)

{

printf("\nThe employee node with the ssn:%s is deleted",first->ssn);

free(first);

count--;

return NULL;

}

prev=NULL;

cur=first;

while(cur->rlink!=NULL)

{

prev=cur;

cur = cur->rlink;

}

cur->llink = NULL;

printf("\nThe employee node with the ssn:%s is deleted",cur->ssn);

free(cur);

prev->rlink = NULL;

count--;

return first;

}

void deqdemo()

{

int ch;

while(1)

{

printf("\nDemo Double Ended Queue Operation");

printf("\n1:InsertQueueFront\n 2: DeleteQueueFront\n 3:InsertQueueRear\n 4:DeleteQueueRear\n 5:DisplayStatus\n 6: Exit \n");

scanf("%d", &ch);

switch(ch)

{

case 1: first=insertfront();

break;

case 2: first=deletefront();

break;

case 3: first=insertend();

break;

case 4: first=deleteend();

break;

case 5: display();

break;

default : return;

}

}

}

void main()

{

int ch,i,n;

while(1)

{

printf("\n\n~~~Menu~~~");

printf("\n1:Create DLL of Employee Nodes");

printf("\n2:DisplayStatus");

printf("\n3:InsertAtEnd");

printf("\n4:DeleteAtEnd");

printf("\n5:InsertAtFront");

printf("\n6:DeleteAtFront");

printf("\n7:Double Ended Queue Demo using DLL");

printf("\n8:Exit \n");

printf("\nPlease enter your choice: ");

scanf("%d",&ch);

switch(ch)

{

case 1 : printf("\nEnter the no of Employees: ");

scanf("%d",&n);

for(i=1;i<=n;i++)

first = insertend();

break;

case 2: display();

break;

case 3: first = insertend();

break;

case 4: first = deleteend();

break;

case 5: first = insertfront();

break;

case 6: first = deletefront();

break;

case 7: deqdemo();

break;

case 8 : exit(0);

default: printf("\nPlease Enter the valid choice");

}

}

}

Output:

~~~Menu~~~

1:Create DLL of Employee Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:InsertAtFront

6:DeleteAtFront

7:Double Ended Queue Demo using DLL

8:Exit

Please enter your choice: 1

Enter the no of Employees: 2

Enter the ssn,Name,Department,Designation,Salary,PhoneNo of the employee:

111

aaa

dept1

des1

1000

11111

Enter the ssn,Name,Department,Designation,Salary,PhoneNo of the employee:

222

bbb

dept2

des2

2000

22222

~~~Menu~~~

1:Create DLL of Employee Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:InsertAtFront

6:DeleteAtFront

7:Double Ended Queue Demo using DLL

8:Exit

Please enter your choice: 2

ENode:1||SSN:111|Name:aaa|Department:dept1|Designation:des1|Salary:1000|Phone no:11111

ENode:2||SSN:222|Name:bbb|Department:dept2|Designation:des2|Salary:2000|Phone no:22222

No of employee nodes is 2

~~~Menu~~~

1:Create DLL of Employee Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:InsertAtFront

6:DeleteAtFront

7:Double Ended Queue Demo using DLL

8:Exit

Please enter your choice: 3

Enter the ssn,Name,Department,Designation,Salary,PhoneNo of the employee:

333

ccc

dept3

des3

3000

33333

~~~Menu~~~

1:Create DLL of Employee Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:InsertAtFront

6:DeleteAtFront

7:Double Ended Queue Demo using DLL

8:Exit

Please enter your choice: 2

ENode:1||SSN:111|Name:aaa|Department:dept1|Designation:des1|Salary:1000|Phone no:11111

ENode:2||SSN:222|Name:bbb|Department:dept2|Designation:des2|Salary:2000|Phone no:22222

ENode:3||SSN:333|Name:ccc|Department:dept3|Designation:des3|Salary:3000|Phone no:33333

No of employee nodes is 3

~~~Menu~~~

1:Create DLL of Employee Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:InsertAtFront

6:DeleteAtFront

7:Double Ended Queue Demo using DLL

8:Exit

Please enter your choice: 5

Enter the ssn,Name,Department,Designation,Salary,PhoneNo of the employee:

444

ddd

dept4

des4

4000

44444

~~~Menu~~~

1:Create DLL of Employee Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:InsertAtFront

6:DeleteAtFront

7:Double Ended Queue Demo using DLL

8:Exit

Please enter your choice: 2

ENode:1||SSN:444|Name:ddd|Department:dept4|Designation:des4|Salary:4000|Phone no:44444

ENode:2||SSN:111|Name:aaa|Department:dept1|Designation:des1|Salary:1000|Phone no:11111

ENode:3||SSN:222|Name:bbb|Department:dept2|Designation:des2|Salary:2000|Phone no:22222

ENode:4||SSN:333|Name:ccc|Department:dept3|Designation:des3|Salary:3000|Phone no:33333

No of employee nodes is 4

~~~Menu~~~

1:Create DLL of Employee Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:InsertAtFront

6:DeleteAtFront

7:Double Ended Queue Demo using DLL

8:Exit

Please enter your choice: 4

The employee node with the ssn:333 is deleted

~~~Menu~~~

1:Create DLL of Employee Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:InsertAtFront

6:DeleteAtFront

7:Double Ended Queue Demo using DLL

8:Exit

Please enter your choice: 6

The employee node with the ssn:444 is deleted

~~~Menu~~~

1:Create DLL of Employee Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:InsertAtFront

6:DeleteAtFront

7:Double Ended Queue Demo using DLL

8:Exit

Please enter your choice: 2

ENode:1||SSN:111|Name:aaa|Department:dept1|Designation:des1|Salary:1000|Phone no:11111

ENode:2||SSN:222|Name:bbb|Department:dept2|Designation:des2|Salary:2000|Phone no:22222

No of employee nodes is 2

~~~Menu~~~

1:Create DLL of Employee Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:InsertAtFront

6:DeleteAtFront

7:Double Ended Queue Demo using DLL

8:Exit

Please enter your choice: 7

Demo Double Ended Queue Operation

1:InsertQueueFront

2: DeleteQueueFront

3:InsertQueueRear

4:DeleteQueueRear

5:DisplayStatus

6: Exit

2

The employee node with the ssn:111 is deleted

Demo Double Ended Queue Operation

1:InsertQueueFront

2: DeleteQueueFront

3:InsertQueueRear

4:DeleteQueueRear

5:DisplayStatus

6: Exit

4

The employee node with the ssn:222 is deleted

Demo Double Ended Queue Operation

1:InsertQueueFront

2: DeleteQueueFront

3:InsertQueueRear

4:DeleteQueueRear

5:DisplayStatus

6: Exit

2

Doubly Linked List is empty

Demo Double Ended Queue Operation

1:InsertQueueFront

2: DeleteQueueFront

3:InsertQueueRear

4:DeleteQueueRear

5:DisplayStatus

6: Exit

6

~~~Menu~~~

1:Create DLL of Employee Nodes

2:DisplayStatus

3:InsertAtEnd

4:DeleteAtEnd

5:InsertAtFront

6:DeleteAtFront

7:Double Ended Queue Demo using DLL

8:Exit

Please enter your choice: 8