Experiment-2.4

A menu driven Program for the operations on Doubly Linked List (DLL).

Student Name: Neha Sharma **UID: 20BCS4576**
Branch: 20IOT1 **Section/Group: A**
Semester: 3rd **Date of Performance:05.10.21**
Subject Name: DATA STRUCTURES LAB **Subject Code:** 21O-20CSP-236_20BIT-1_B

1. Aim/Overview of the practical:

A menu driven Program for the operations on Doubly Linked List (DLL).

2. Task to be done:

We have to do different operation on Doubly Linked List (DLL).

3. Algorithm/Flowchart:

Step 1: Start.
Step 2: Read the value of N. (N student's information)
Step 3: Create a doubly linked list. (DLL)
Step 4: Display the status of DLL.
Step 5: Count the number of nodes.
Step 6: Perform insertion at front of list.
Step 7: Perform deletion at the front of the list.
Step 8: Perform insertion at end of the list.
Step 9: Perform deletion at the end of the list
Step 10: Demonstrate how doubly linked list can be used as double ended queue.
Step 11: Stop.

4. Code for experiment/practical:

```
#include<stdio.h>
#include<conio.h>
```

```
int MAX=4, count;
struct emp
{
int ssn;
char name[20];
char dept[10];
char desig[15];
int sal;
char phno[10];
struct emp *left;
struct emp *right;
};
typedef struct emp NODE;
int countnodes(NODE *head)
{
NODE *p;
count=0;
p=head;
while(p!=NULL)
{
p=p->right;
count++;
}
return count;
}
NODE* getnode(NODE *head)
{
NODE *newnode;
newnode=(NODE*)malloc(sizeof(NODE));
newnode->right=newnode->left=NULL;
printf("\nEnter SSN, Name, Dept, Designation, Sal, Ph.No\n");
scanf("%d",&newnode->ssn);
flushall();
gets(newnode->name);
flushall();

gets(newnode->dept);
flushall();
gets(newnode->desig);
scanf("%d",&newnode->sal);
```

```
flushall();
gets(newnode->phno);
head=newnode;
return head;
}
NODE* display(NODE *head)
{
NODE *p;
if(head==NULL)
printf("\nNo Employee data\n");
else
{
p=head;
printf("\n----EMPLOYEE DATA----\n");
printf("\nSSN\tNAME\tDEPT\tDESINGATION\tSAL\t\tPh.NO.");
while(p!=NULL)
{
printf("\n%d\t%s\t%s\t%s\t\t%d\t\t%s", p->ssn, p->name, p->dept, p->desig,
p->sal, p->phno);
p = p->right; //Go to next node...
}
printf("\nThe no. of nodes in list is: %d",countnodes(head));
}
return head;
}
NODE* create(NODE *head)// creating & inserting at end.
{
NODE *p, *newnode;
p=head;
if(head==NULL)
{
newnode=getnode(head);
head=newnode;
}
else
{
newnode=getnode(head);
while(p->right!=NULL)
{
```

```
p=p->right;
}
p->right=newnode;
newnode->left=p;
}
return head;
}
NODE* insert_end(NODE *head)
{
if(countnodes(head)==MAX)
printf("\nList is Full!!");
else
head=create(head);
return head;
}
NODE* insert_front(NODE *head)
{
NODE *p, *newnode;
if(countnodes(head)==MAX)
printf("\nList is Full!!");
else
{
if(head==NULL)
{
newnode=getnode(head);
head=newnode; //set first node to be head
}
else
{
}
}
return head;
}
newnode=getnode(head);
newnode->right=head;
head->left=newnode;
head=newnode;

}
}
```

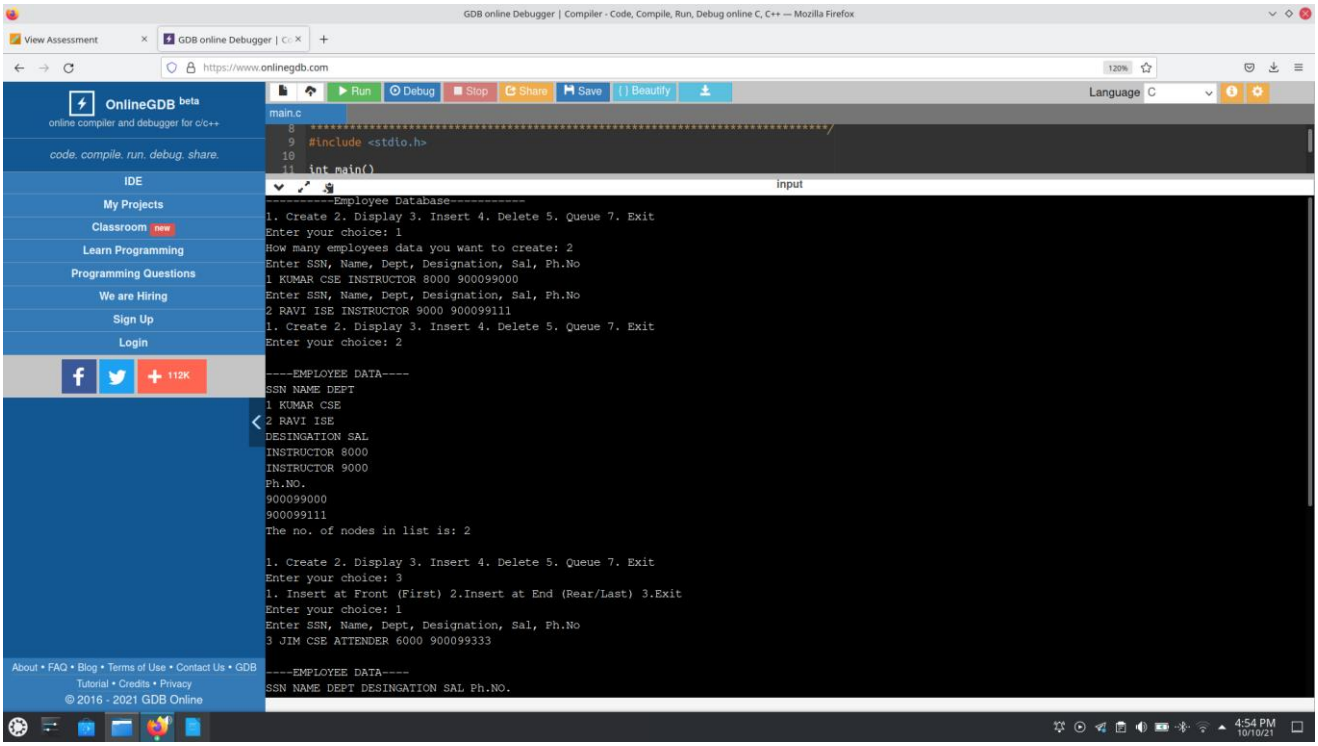
```
return head;
}

NODE* insert(NODE *head)
{
int ch;
do
{
printf("\n 1.Insert at Front(First) \t 2.Insert at
End(Rear/Last)\t3.Exit");
printf("\nEnter your choice: ");
scanf("%d", &ch);
switch(ch)
{
case 1: head=insert_front(head); break;
case 2: head=insert_end(head); break;
case 3: break;
}
head=display(head);
}while(ch!=3);
return head;
}
NODE* delete_front(NODE *head)
{
NODE *p;
if(head==NULL)
printf("\nList is Empty (QUEUE)");
else
{
p=head;
head=head->right;
head->right->left=NULL;
free(p);
printf("\nFront(first)node is deleted");
}
return head;
}
NODE* delete_end(NODE *head)
{
NODE *p, *q;
```

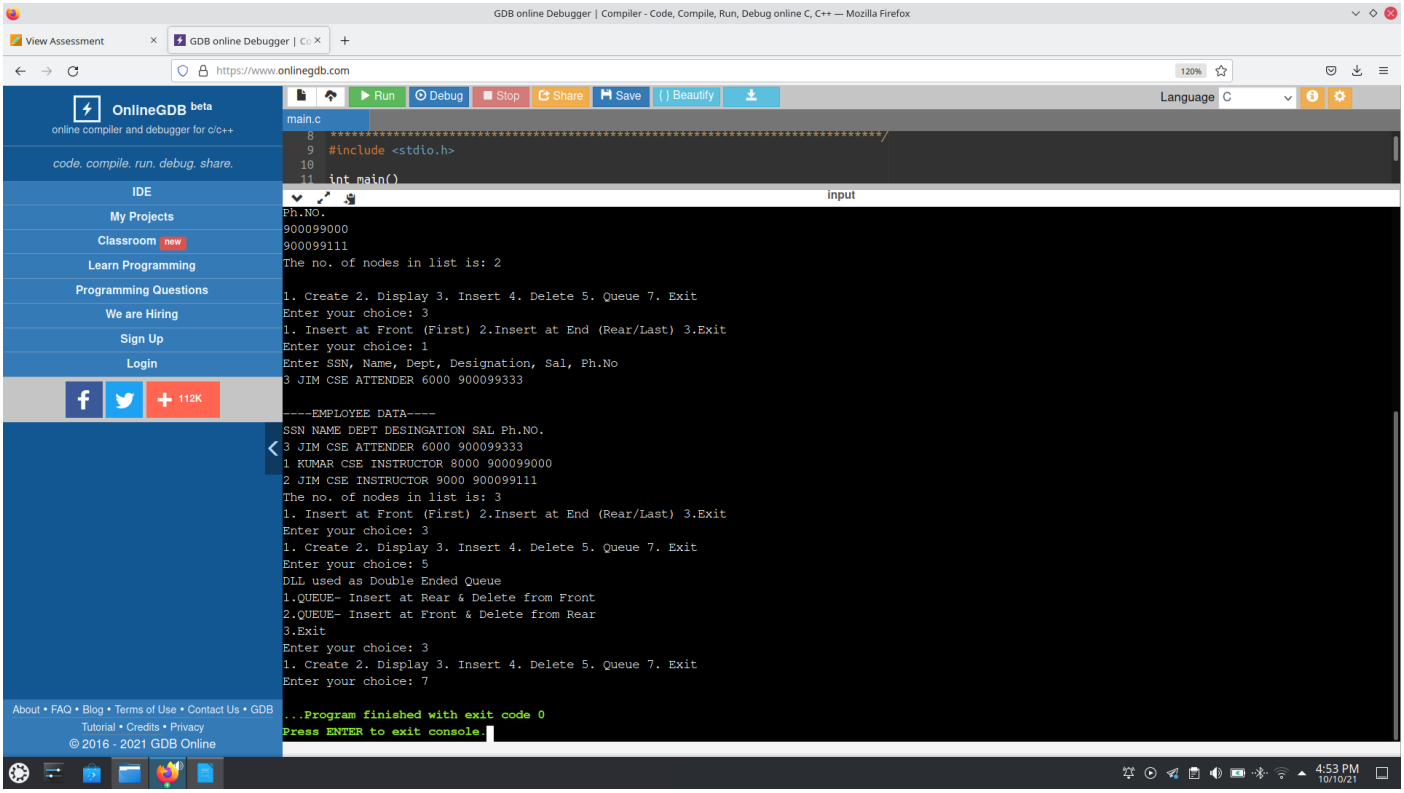
```
p=head;
while(p->right!=NULL)
{
p=p->right; //Go upto -1 node which you want to delete
}
q=p->left;

q->right=NULL;
p->left=NULL;
free(p);//Delete last node...
printf("\nLast(end) entry is deleted");
return head;
}
NODE *del(NODE *head)
{
int ch;
do {
printf("\n1.Delete from Front(First)\t2. Delete from
End(Rear/Last))\t3.Exit");
printf("\nEnter your choice: ");
scanf("%d", &ch);
switch(ch)
{
case 1: head=delete_front(head);
break;
case 2: head=delete_end(head);
break;
case 3: break;
}
head=display(head);
}while(ch!=3);
return head;
}
NODE* queue(NODE *head)
{
int ch, ch1, ch2;
do
{
printf("\nDLL used as Double Ended Queue");
printf("\n1.QUEUE- Insert at Rear & Delete from Front");
```

```
printf("\n2.QUEUE- Insert at Front & Delete from Rear");
printf("\n3.Exit");
printf("\nEnter your choice: ");
scanf("%d", &ch);
switch(ch)
{
case 1: do{
printf("\n1.Insert at Rear\t2.Delete from From Front\t3.Exit");
printf("\nEnter your choice: ");
scanf("%d", &ch1);
switch(ch1)

{
case 1: head=insert_end(head); break;
case 2: head=delete_front(head); break;
case 3: break;
}
}while(ch1!=3);
break;
case 2: do{
printf("\n1.Insert at Front\t2.Delete from Rear\t3.Exit");
printf("\nEnter your choice: ");
scanf("%d", &ch2);
switch(ch2)
{
case 1: head=insert_front(head); break;
case 2: head=delete_end(head); break;
case 3: break;
}
}while(ch2!=3);
break;
case 3: break;
}
}while(ch!=3);
head=display(head);
return head;
}
void main()
{
int ch, i, n;
```

```
NODE *head;
head=NULL;
clrscr();
printf("\n----------Employee Database----------");
do
{
printf("\n1.Create\t2.Display\t3.Insert\t4.Delete\t5.Queue\t6.Exit");
printf("\nEnter your choice: ");
scanf("%d", &ch);
switch(ch)
{
case 1: printf("\nHow many employees data you want to create: ");
scanf("%d", &n);
for(i=0;i<n;i++)
head=create(head);//Call to Create node...
break;
case 2: head=display(head); //Call to Display...
break;

case 3: head=insert(head); //Call to Insert...
break;
case 4: head=del(head); //Call to delete
break;
case 5: head=queue(head);
break;
case 6: exit(0); //Exit...
break;
}
}while(ch!=6);
}
```

5. Output: Image of sample output to be attached here


GDB online Debugger | Compiler - Code, Compile, Run, Debug online C, C++ — Mozilla Firefox
https://www.onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
We are Hiring
Sign Up
Login
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
#include <stdio.h>
int main()
input
----------Employee Database----------
1. Create 2. Display 3. Insert 4. Delete 5. Queue 7. Exit
Enter your choice: 1
How many employees data you want to create: 2
Enter SSN, Name, Dept, Designation, Sal, Ph.No
1 KUMAR CSE INSTRUCTOR 8000 900099000
Enter SSN, Name, Dept, Designation, Sal, Ph.No
2 RAVI ISE INSTRUCTOR 9000 900099111
1. Create 2. Display 3. Insert 4. Delete 5. Queue 7. Exit
Enter your choice: 2
----EMPLOYEE DATA----
SSN NAME DEPT
1 KUMAR CSE
2 RAVI ISE
DESINGATION SAL
INSTRUCTOR 8000
INSTRUCTOR 9000
Ph.NO.
900099000
900099111
The no. of nodes in list is: 2
1. Create 2. Display 3. Insert 4. Delete 5. Queue 7. Exit
Enter your choice: 3
1. Insert at Front (First) 2.Insert at End (Rear/Last) 3.Exit
Enter your choice: 1
Enter SSN, Name, Dept, Designation, Sal, Ph.No
3 JIM CSE ATTENDER 6000 900099333
----EMPLOYEE DATA----
SSN NAME DEPT DESINGATION SAL Ph.NO.
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online
4:54 PM
10/10/21


GDB online Debugger | Compiler - Code, Compile, Run, Debug online C, C++ — Mozilla Firefox
https://www.onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
We are Hiring
Sign Up
Login
Run
Debug
Stop
Share
Save
{ } Beautify
Language C
main.c
#include <stdio.h>
int main()
input
Ph.NO.
900099000
900099111
The no. of nodes in list is: 2
1. Create 2. Display 3. Insert 4. Delete 5. Queue 7. Exit
Enter your choice: 3
1. Insert at Front (First) 2.Insert at End (Rear/Last) 3.Exit
Enter your choice: 1
Enter SSN, Name, Dept, Designation, Sal, Ph.No
3 JIM CSE ATTENDER 6000 900099333
----EMPLOYEE DATA----
SSN NAME DEPT DESINGATION SAL Ph.NO.
3 JIM CSE ATTENDER 6000 900099333
1 KUMAR CSE INSTRUCTOR 8000 900099000
2 JIM CSE INSTRUCTOR 9000 900099111
The no. of nodes in list is: 3
1. Insert at Front (First) 2.Insert at End (Rear/Last) 3.Exit
Enter your choice: 3
1. Create 2. Display 3. Insert 4. Delete 5. Queue 7. Exit
Enter your choice: 5
DLL used as Double Ended Queue
1.QUEUE- Insert at Rear & Delete from Front
2.QUEUE- Insert at Front & Delete from Rear
3.Exit
Enter your choice: 3
1. Create 2. Display 3. Insert 4. Delete 5. Queue 7. Exit
Enter your choice: 7
...Program finished with exit code 0
Press ENTER to exit console.
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2021 GDB Online
4:53 PM
10/10/21

Learning outcomes (What I have learnt):

- *Add item to the Doubly Linked List (DLL).*
- *Remove an item from the Doubly Linked List (DLL)..*
- *Evaluation of expressions.*
- *Backtracking.*
- *Runtime memory management.*

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			