

Lab program 5:

5) Write a program to implement Singly linked list with following operators : a> Create an linked list , b> Insertion of a node at first position, at any position and at end of list . c> Display the contents of linked list .

A)

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

struct node
{
    int info;
    struct node *link;
};

typedef struct node *NODE;

NODE getnode()
{
    NODE x;
    x = (NODE)malloc(sizeof(struct node));
    if (x == NULL)
    {
        printf("mem full \n");
        exit(0);
    }
    return x;
}

NODE insert_front(NODE first, int item)
{
    NODE temp;
    temp = getnode();
    temp->info = item;
    temp->link = NULL;
}
```

```
    NODE insert_pos(NODE first, int item, int pos)
{
    NODE temp;
    temp = getnode();
    temp->info = item;
    temp->link = NULL;
```

```
if (first == NULL) {  
    return temp;  
    temp->link = first;  
    first = temp;  
    return first;  
}
```

```
} NODE insert_incar(NODE first, int item){
```

```
    NODE temp, cur;  
    temp = getnode();  
    temp->info = item;  
    temp->link = NULL;  
    if (first == NULL)  
        return temp;  
    cur = first;  
    while (cur->link != NULL)  
        cur = cur->link;  
    cur->link = temp;  
    return first;
```

```
}
```

```
void display(NODE first){
```

```
    NODE temp;  
    if (first == NULL)  
        printf("list empty cannot display items \n");  
    for (temp = first; temp != NULL; temp = temp->link) {  
        printf("%d \n", temp->info);
```

```
}
```

```
} NODE insert_pos(int item, int pos, NODE first){
```

```
    NODE temp;  
    NODE prev, cur;  
    int count;  
    temp = getnode();  
    temp->info = item;  
    temp->link = NULL;
```

```
if(first == NULL && pos == 1) {
```

```
    return temp;
```

{

```
if(first == NULL) {
```

```
    printf("invalid position\n");
```

```
    return first;
```

}

```
if(pos == 1) {
```

```
    temp->link = first;
```

```
    return temp;
```

?

```
count = 1;
```

```
prev = NULL;
```

```
cur = first;
```

```
while(cur != NULL && count != pos) {
```

```
    prev = cur;
```

```
    cur = cur->link;
```

```
    count++;
```

?

```
if(count == pos) {
```

```
    prev->link = temp;
```

```
    temp->link = cur;
```

```
    return first;
```

?

```
printf("invalid position\n");
```

```
return first;
```

?

```
int main() {
```

```
    int item, choice, pos;
```

```
    NODE first = NULL;
```

```
    system("cls");
```

```
    for(;;) {
```

```
        printf("\n 1: Insert front\n 2: Insert at specified  
position\n 3: Insert rear\n 4: Display list\n 5: Exit
```

in");

~~scanf("%d", &choice);~~

printf("enter the choice \n");

scanf("%d", &choice);

switch(choice) {

case 1: printf("enter the item at front-end : \n");

scanf("%d", &item);

first = insert_front(first, item);

break;

case 2: printf("enter the item to be inserted: \n");

scanf("%d", &item);

printf("enter the position at which item to be inserted: \n");

scanf("%d", &pos);

first = insert_pos(item, pos, first);

break;

case 3: printf("enter the item at rear-end \n");

scanf("%d", &item);

first = insert_rear(first, item);

break;

case 4: display(first);

break;

default: exit(0);

break;

}

getch();

return 0;

}

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list insert\singly_linked_lists_insertion.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
1 #include<stdio.h>
2 #include<conio.h>
3 #include<malloc.h>
4 #include<process.h>
5 struct node
6 {
7     int info;
8     struct node *link;
9 };
10 typedef struct node *NODE;
11 NODE get_node()
12 {
13     NODE x;
14     x=(NODE) malloc(sizeof(struct node));
15     if(x==NULL)
16     {
17         printf("mem full\n");
18         exit(0);
19     }
20     return x;
21 }
22 int freenode(NODE x)
23 {
24     free(x);
25     return 0;
26 }
27 NODE insert_front(NODE first, int item)
28 {
29     NODE temp;
30     temp=get_node();
31     temp->info=item;
32     temp->link=NULL;
33     if(first==NULL)
34         return temp;
35     temp->link=first;
36     first=temp;
37     return first;
38 }
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 18 Sel: 0 Lines: 139 Length: 2287 Insert Done parsing in 0.016 seconds

Search

Windows Start File Explorer Mail Settings Task View Chrome Edge Word DEV

20:43 ENG 05-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list insert\singly_linked_lists_insertion.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
38 L }
39
40 NODE insert_rear(NODE first, int item)
41 {
42     NODE temp, cur;
43     temp = get_node();
44     temp->info = item;
45     temp->link = NULL;
46     if(first == NULL)
47         return temp;
48     cur = first;
49     while(cur->link != NULL)
50         cur = cur->link;
51     cur->link = temp;
52     return first;
53 }
54
55 void display(NODE first)
56 {
57     NODE temp;
58     if(first == NULL)
59         printf("list empty cannot display items\n");
60     for(temp = first; temp != NULL; temp = temp->link)
61     {
62         printf("%d\n", temp->info);
63     }
64
65 NODE insert_pos( int item, int pos, NODE first )
66 {
67     NODE temp;
68     NODE prev, cur;
69     int count;
70     temp = get_node();
71     temp->info = item;
72     temp->link = NULL;
73     if(first == NULL && pos == 1)
74     {
75         return temp;
76     }
77     else
78     {
79         cur = first;
80         prev = NULL;
81         count = 1;
82         while(count < pos && cur != NULL)
83         {
84             prev = cur;
85             cur = cur->link;
86             count++;
87         }
88         if(cur == NULL)
89             return NULL;
90         temp->link = cur;
91         if(prev == NULL)
92             first = temp;
93         else
94             prev->link = temp;
95     }
96     return first;
97 }
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 18 Sel: 0 Lines: 139 Length: 2287 Insert Done parsing in 0.016 seconds

Search

20:43 05-12-2020 ENG

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list insert\singly_linked_lists_insertion.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
singly_linked_lists_insertion.cpp
75     return temp;
76 }
77 if(first==NULL)
78 {
79     printf("invalid position \n");
80     return first;
81 }
82 if(pos==1)
83 {
84     temp->link=first;
85     return temp;
86 }
87 count=1;
88 prev=NULL;
89 cur=first;
90 while(cur!=NULL && count!=pos)
91 {
92     prev=cur;
93     cur=cur->link;
94     count++;
95 }
96 if(count==pos)
97 {
98     prev->link=temp;
99     temp->link=cur;
100    return first;
101 }
102 printf("invalid position \n");
103 return first;
104 }
105 int main()
106 {
107     int item,choice,pos;
108     NODE first=NULL;
109     system("cls");
110     for(;;)
111     {
112         printf("\n 1:Insert front\n 2:Insert at specified position \n 3:Insert rear\n 4:Display list\n 6:Exit\n");
113         choice=getchar();
114         switch(choice)
115         {
116             case '1':
117                 insert_at_head(&first);
118                 break;
119             case '2':
120                 insert_at_pos(&first,&item,&pos);
121                 break;
122             case '3':
123                 insert_at_rear(&first,&item);
124                 break;
125             case '4':
126                 display_list(first);
127                 break;
128             case '6':
129                 exit(0);
130         }
131     }
132 }
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 18 Sel: 0 Lines: 139 Length: 2287 Insert Done parsing in 0.016 seconds

Search

20:43 05-12-2020 ENG

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list insert\singly_linked_lists_insertion.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
singly_linked_lists_insertion.cpp
103     return first;
104 }
105 int main()
106 {
107     int item_choice, pos;
108     NODE first=NULL;
109     system("cls");
110     for(;;)
111     {
112         printf("\n 1:Insert_front\n 2:Insert at specified position \n 3:Insert_rear\n 4:Display_list\n 6:Exit\n");
113         printf("enter the choice\n");
114         scanf ("%d", &choice);
115         switch(choice)
116         {
117             case 1: printf("enter the item at front-end\n");
118             scanf ("%d", &item);
119             first=insert_front(first,item);
120             break;
121             case 2: printf("enter the item to be inserted:\n");
122             scanf ("%d", &item);
123             printf("enter the position at which item to be inserted:\n");
124             scanf ("%d", &pos);
125             first=insert_pos(item,pos,first);
126             break;
127             case 3: printf("enter the item at rear-end\n");
128             scanf ("%d", &item);
129             first=insert_rear(first,item);
130             break;
131             case 4: display(first);
132             break;
133             default: exit(0);
134             break;
135         }
136     }
137     getch();
138     return 0;
139 }
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 18 Sel: 0 Lines: 139 Length: 2287 Insert Done parsing in 0.016 seconds

Search

20:43 05-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list insert\singly_linked_lists_insertion.exe

```
1:Insert_front  
2:Insert at specified position  
3:Insert_rear  
4:Display_list  
6:Exit
```

enter the choice

1
enter the item at front-end
54

```
1:Insert_front  
2:Insert at specified position  
3:Insert_rear  
4:Display_list  
6:Exit
```

enter the choice

3
enter the item at rear-end

78

```
1:Insert_front  
2:Insert at specified position  
3:Insert_rear  
4:Display_list  
6:Exit
```

enter the choice

2
enter the item to be inserted:

65
enter the position at which item to be inserted:

3

```
1:Insert_front  
2:Insert at specified position  
3:Insert_rear  
4:Display_list  
6:Exit
```

enter the choice

2
enter the item to be inserted:

1
enter the position at which item to be inserted:

2

```
1:Insert_front  
2:Insert at specified position  
3:Insert_rear  
4:Display_list  
6:Exit
```

enter the choice

1
enter the item to be inserted:

1
enter the position at which item to be inserted:

2

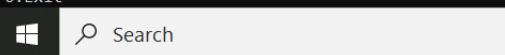
```
1:Insert_front  
2:Insert at specified position  
3:Insert_rear  
4:Display_list  
6:Exit
```

enter the choice

1
enter the item to be inserted:

1
enter the position at which item to be inserted:

2



C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list insert\singly_linked_lists_insertion.exe

78

1:Insert_front
2:Insert at specified position

3:Insert_rear
4:Display_list
6:Exit

enter the choice

2

enter the item to be inserted:

65

enter the position at which item to be inserted:

3

1:Insert_front
2:Insert at specified position

3:Insert_rear
4:Display_list
6:Exit

enter the choice

2

enter the item to be inserted:

1

enter the position at which item to be inserted:

2

1:Insert_front
2:Insert at specified position

3:Insert_rear
4:Display_list
6:Exit

enter the choice

4

54

1

78

65

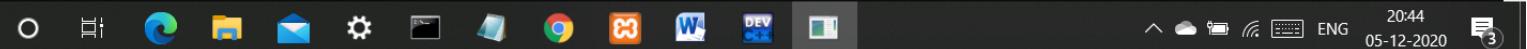
1:Insert_front
2:Insert at specified position

3:Insert_rear
4:Display_list
6:Exit

enter the choice



Search



20:44
ENG
05-12-2020
3