, 20	classmate
	S. R. Poo JA
	1BM19CS135
	# include < stdro. h)
	#include < stdlib.h>
	#include < mallocin>
	skuct node
	E (inthering) is not be interested in its
	struct node *clink; 3',
	type of skuck node * NODE;
	NODE getnodel) &
	E NODE X's seed to note the solo ") Have a
	2. (NODE) malor (szeg (skurt node)),
	of (n== NULL) Thord mad to react towns = -120)
	5 in the state of
	penty (umen full n'i) sel tent del la des les sens
	ext (0), &
	Return of 3 (hours stollable of the 14 secon
	void feanale CNODE x)
	E fee (x) " +/ tonds par the very Hray is sen
	g (motion "b. rullogs
	NODE insert_was (NODE great intitlem)
200	c rote temp and the man and all the control of the
	targe gernauch
petel	temp -> into = item;
	temp-> link = NULL,
-	J (dist = 1 NOCC)
-	Selven tens
	cup fust of the solar m) Have the sens
	while (cue > link!= pvul)
	cur- cur > link; selius furt, 3
	LARDE delike 1 + Court 15 12
	Relian first', 3 WODE delete-first (NODE first) ? NODE temp'
	if (first == NULL) {
	point (" list is emply cannot delete \n");
	Xeliun foxe, I
	temp= first;

3	CO Proto
	S. R. Poota IBMI9 CS135
	101/1(7(2)12)
	int main()?
lo ?	int stem choice, pos'
	NODE tast=NUU,
	Sustem ("cli")
	12 (11)
	1 pent ("In Queus operations", " " " " " " " Letter")
	In 3. Died on list property of a state of the set
	In 5: Delete Lent In 6: Dis play list (Stack) in a course
	pent (" enter the choice laid);
	sear ("v.d", schoice);
	switch (choice) of (all tool and to the series)
	E 3 (most the test front from 3000)
	Case 1: prenty ("enter the item at seas end In");
	Scarl ("7.d", & item); (Inhante og not
	flest = Meet- eas (first, item),
	break's will- will- will-
	Case 2' fiest = delete-funt (fiest),
	Ireak'
	case 36: display (fixst); 700h) & that states 3001
	break;
	case 4: pents ("enter the iten at front end \n");
	scant ("Y.d", & item); fiest = insert-front (fiest, Ptem);
	fiest = insert-floot (fiest, tem)
	skeak)
	case 5: fiest = delete-funt _ s (fiest)',
ada	break; at his day to hold not a fing
	case 6' display-8 (first);
	break;
	default & ! exit (o);
	3001
	3 "Charlements maked trans with as a men were
	getch()
	Return O',
	3 88 (doce gross (4,16,10)) have

```
1
     #include<stdio.h>
2
     #include<stdlib.h>
3
     #include<malloc.h>
4
     struct node
5
     {
6
    int info:
7
     struct node *link;
8
     }:
9
     typedef struct node *NODE;
     NODE getnode()
10
11
     {
12
     NODE x;
13
     x=(NODE)malloc(sizeof(struct node));
14
     if(x==NULL)
15
16
     printf("mem full\n");
17
     exit(0);
18
     }
19
     return x;
20
     }
21
     void freenode(NODE x)
22
     {
23
     free(x);
24
25
     NODE insert_rear(NODE first,int item)
26
     {
27
     NODE temp, cur;
28
     temp=getnode();
29
     temp->info=item;
30
     temp->link=NULL;
    if(first==NULL)
31
32
    return temp;
33
     cur=first;
34
    while(cur->link!=NULL)
35
     cur=cur->link;
36
     cur->link=temp;
37
     return first;
38
     }
39
     NODE delete_front(NODE first)
40
     {
41
     NODE temp;
42
     if(first==NULL)
43
44
     printf("list is empty cannot delete\n");
```

```
45
     return first;
46
     temp=first:
47
     temp=temp->link;
48
49
     printf("item deleted at front-end is=%d\n",first->info);
50
     free(first);
     return temp;
51
52
     }
     void display(NODE first)
53
54
55
     NODE temp;
56
     if(first==NULL)
     printf("list empty cannot display items\n");
57
     for(temp=first;temp!=NULL;temp=temp->link)
58
59
60
     printf("%d \n", temp->info);
61
     }
62
     }
63
     NODE insert_front(NODE first,int item)
64
     {
65
     NODE temp;
     temp=getnode();
66
67
     temp->info=item;
68
     temp->link=NULL;
69
     if(first==NULL)
70
     return temp;
71
     temp->link=first;
72
     first=temp;
73
     return first;
74
75
     NODE delete_front_s(NODE first)
76
     {
77
     NODE temp;
78
     if(first==NULL)
79
     printf("stack is empty cannot delete\n");
80
     return first;
81
82
     }
83
     temp=first;
84
     temp=temp->link;
85
     printf("item deleted at front-end is=%d\n", first->info);
86
     free(first);
87
     return temp;
22
     ı
```

```
92
      if(first==NULL)
      printf("stack empty cannot display items\n");
 93
      for(temp=first;temp!=NULL;temp=temp->link)
 94
 95
 96
      printf("%d\n", temp->info);
 97
 98
      }
 99
      int main()
100
      int item, choice, pos;
101
102
      NODE first=NULL;
103
      system("cls");
104
      for(::)
105
106
      printf("\n Queue operations :\n 1:Insert_rear\n 2:Delete_front\n
      3:Display_list(Queue)\n \n Stack operations \n 4:Insert_front\n 5:
      Delete_front \n 6:Dislay_list(Stack)\n 7:Exit \n \n");
107
      printf("enter the choice \n");
      scanf("%d",&choice);
108
109
      switch(choice)
110
111
      case 1:printf("enter the item at rear-end\n");
      scanf("%d",&item);
112
      first=insert_rear(first,item);
113
114
      break:
      case 2:first=delete_front(first);
115
116
      break:
117
      case 3:display(first);
118
      break:
      case 4:printf("enter the item at front-end\n");
119
120
      scanf("%d",&item);
```

```
> clang-7 -pthread -lm -o main main.c
                                                                               Qem
main.c:131:1: warning: implicit declaration of function 'getch' is invalid in
      C99 [-Wimplicit-function-declaration]
getch();
1 warning generated.
./main
sh: 1: cls: not found
 Queue operations :
 1: Insert_rear
 2:Delete_front
 3:Display_list(Queue)
 Stack operations
 4: Insert_front
 5: Delete_front
 6:Dislay_list(Stack)
 7:Exit
enter the choice
enter the item at rear-end
23
 Queue operations :
 1:Insert_rear
 2:Delete_front
 3:Display_list(Queue)
 Stack operations
 4: Insert_front
 5: Delete_front
 6:Dislay_list(Stack)
 7:Exit
enter the choice
enter the item at rear-end
12
```

```
enter the choice
                                                                                 Q es
enter the item at rear-end
12
Queue operations :
 1: Insert_rear
 2:Delete_front
 3:Display_list(Queue)
 Stack operations
4: Insert_front
 5: Delete_front
6:Dislay_list(Stack)
 7:Exit
enter the choice
2
item deleted at front-end is=23
 Queue operations :
 1: Insert_rear
 2:Delete_front
 3:Display_list(Queue)
 Stack operations
 4: Insert_front
 5: Delete_front
6:Dislay_list(Stack)
 7:Exit
enter the choice
3
12
 Queue operations :
 1:Insert_rear
 2:Delete_front
 3:Display_list(Queue)
 Stack operations
 4: Insert_front
 5: Delete_front
6:Dislay_list(Stack)
 7:Exit
```

```
enter the item at front-end
                                                                                Q
14
 Queue operations :
 1:Insert_rear
 2:Delete_front
 3:Display_list(Queue)
 Stack operations
 4: Insert_front
 5: Delete_front
6:Dislay_list(Stack)
 7:Exit
enter the choice
enter the item at front-end
17
 Queue operations :
 1:Insert_rear
 2:Delete_front
 3:Display_list(Queue)
Stack operations
 4: Insert_front
 5: Delete_front
6:Dislay_list(Stack)
 7:Exit
enter the choice
6
17
14
Queue operations :
 1: Insert_rear
 2:Delete_front
 3:Display_list(Queue)
 Stack operations
 4: Insert_front
 5: Delete_front
6:Dislay_list(Stack)
 7:Exit
enter the choice
```