

```

1  #include <iostream>
2  #include <queue>
3  #include <stack>
4  #include <cstdlib>
5  using namespace std;
6  int select;
7  int op;
8  int op1;
9  int val;
10 void main_maseg()
11 {
12     cout<<"What Do You Want To Represent? \n\n";
13     cout<<" 1: Stack \n";
14     cout<<" 2: Queue \n";
15     cout<<" 3: Linked List \n";
16     cout<<" 4: Double List \n\n";
17     cout<<"Enter The Transaction Number-> ";
18 }
19 ///////////////////////////////////////////////////liner queue////////////////////////////////////
20 struct liner_q
21 {
22     int liner_q_data;
23     liner_q *liner_q_next;
24 };
25 liner_q *liner_q_head=NULL,*liner_q_tail=NULL,*liner_q_np=NULL;
26 int liner_q_isempty()
27 {
28     if (liner_q_head == NULL)
29         return 0;
30     else
31         return 1;
32 }
33 void liner_q_push(int x)
34 {
35     liner_q_np = new liner_q;
36     liner_q_np->liner_q_data = x;
37     liner_q_np->liner_q_next = NULL;
38     if (liner_q_head == NULL)
39         liner_q_head = liner_q_tail = liner_q_np;
40     else
41     {
42         liner_q_tail->liner_q_next = liner_q_np;
43         liner_q_tail = liner_q_np;
44     }
45 }
46 void liner_q_pop()
47 {
48     if (liner_q_isempty())
49     {
50         liner_q *p = liner_q_head;
51         cout <<"delete-> "<<liner_q_head->liner_q_data << endl<< endl;
52         liner_q_head = liner_q_head->liner_q_next;
53         delete p;
54     }
55     else
56         cout << "queue is empty\n";
57     system("pause");
58 }
59 void liner_q_op_menu()
60 {
61     cout<<"    ---Liner Queue---\n";
62     cout<<"Enter the transaction number\n\n";
63     cout<<" 1->Push\n";
64     cout<<" 2->Pop\n\n";
65     cout<<"Press 0 to Main Menu\n\n";
66 }
67 void liner_q_print()
68 {
69     liner_q *temp = liner_q_head;
70     while (temp != NULL)
71     {
72         cout <<"|"<< temp->liner_q_data << "|";
73         temp = temp->liner_q_next;
74     }
75     cout << "\n\n";
76 }
77 ///////////////////////////////////////////////////stack////////////////////////////////////
78 struct stack1
79 {
80     int stack_data;
81     stack1 *stack_last;
82 };
83 stack1 *top=NULL , *stack_np=NULL;
84 int stack_isempty()

```

```

85 {
86     if(top == NULL)
87         return 0;
88     else
89         return 1;
90 }
91 int stack_push (int val)
92 {
93     stack_np=new stack1;
94     stack_np->stack_data = val;
95     stack_np->stack_last = NULL;
96
97     if(top != NULL)
98         stack_np->stack_last = top;
99
100     top = stack_np;
101 }
102 void stack_pop ()
103 {
104     if(stack_isempty())
105     {
106         top = stack_np->stack_last;
107         cout<<"delete-> " <<stack_np->stack_data<<"\n" << endl;
108         delete stack_np;
109         stack_np = top;
110     }
111     else
112         cout<<"stack is empty\n";
113     system("pause");
114 }
115 void stack_print()
116 {
117     stack1 *temp = top;
118     while (temp != NULL)
119     {
120         cout <<"    |" << temp->stack_data << "|\n";
121         temp = temp->stack_last;
122     }
123     cout << "\n\n";
124 }
125
126 void stack_maseg()
127 {
128     cout<<"    ---Stack---\n";
129     cout<<"Enter the transaction number\n\n";
130     cout<<"    1->Push\n";
131     cout<<"    2->Pop\n\n";
132     cout<<"Press 0 to Main Menu\n\n";
133
134 }
135 ///////////////////////////////////////////////////Linked List//////////////////////////////////////
136 struct node
137 {
138     int data;
139     node *next;
140 };
141 node *np=NULL,*head=NULL,*tail=NULL, *l_tail;
142 void create_node()
143 {
144     np=new node;
145     cout<<"input data:";
146     cin>>np->data;
147     np->next=NULL;
148     cout<<"*****\n";
149 }
150 void create_list()
151 {
152     np=new node;
153     cout<<"input data:";
154     cin>>np->data;
155     np->next=NULL;
156     if(head==NULL)
157         head=tail=np;
158     else
159     {
160         tail->next=np;
161         tail=np;
162     }
163 }
164 void print_list()
165 {
166     node *temp = head;
167     while (temp != NULL)
168     {

```

```

169         cout << temp->data << " ";
170         temp = temp->next;
171     }
172     cout << "\n\n";
173 }
174 void add_last()
175 {
176     create_list();
177     tail=head;
178     while(tail->next !=NULL)
179         tail=tail->next;
180
181     tail->next=np;
182     tail=np;
183 }
184 void del_first()
185 {
186     tail=head;
187     head=head->next;
188     delete tail;
189 }
190 void del_last()
191 {
192     tail=head;
193     node *c=NULL;
194     while(tail->next !=NULL)
195     {
196         c=tail;
197         tail=tail->next;
198     }
199     delete tail;
200     c->next=NULL;
201 }
202 void add_first()
203 {
204     create_node();
205     np->next=head;
206     head=np;
207 }
208 int Search_list()
209 {
210     cout<<"input search the num of data:";
211     cin>>val;
212     tail=head;
213     int p=0;
214     while(tail !=NULL)
215     {
216         if(tail->data==val)
217         {
218             p=1;
219             break;
220         }
221         l_tail=tail;
222         tail=tail->next;
223     }
224     return p;
225 }
226 void add_after()
227 {
228     if(Search_list())
229     {
230         create_node();
231         np->next=tail->next;
232         tail->next=np;
233     }
234     else
235         cout<<"not found";
236 }
237 void add_before()
238 {
239     if(Search_list())
240     {
241         if(head==tail)
242         {
243             add_first();
244         }
245         else
246         {
247             create_node();
248             l_tail->next=np;
249             np->next=tail;
250         }
251     }
252     else

```

```

253         cout<<"not found\n";
254     }
255     void del_any()
256     {
257         if(Search_list())
258         {
259             if(tail==head)
260             {
261                 head=head->next;
262                 delete tail;
263             }
264             else
265             {
266                 l_tail->next=tail->next;
267                 delete tail;
268             }
269         }
270         else
271             cout<<"not found\n";
272     }
273     void sort_list()
274     {
275         tail=head;
276         while(tail !=NULL)
277         {
278             node *c=tail->next;
279             while(c!=NULL)
280             {
281                 if(c->data < tail->data)
282                     swap(c->data,tail->data);
283                 c=c->next;
284             }
285             tail=tail->next;
286         }
287     }
288     void linked_list_op menu()
289     {
290         cout<<"    ---Linked List---\n";
291         cout<<"Enter the transaction number\n\n";
292         cout<<" 1->Create list\n";
293         cout<<" 2->Add Last\n";
294         cout<<" 3->Add First\n";
295         cout<<" 4->Add After\n";
296         cout<<" 5->Add Before\n";
297         cout<<" 6->Del First\n";
298         cout<<" 7->Del Last\n";
299         cout<<" 8->Del Any\n\n";
300         cout<<"Press 0 to Main Menu\n\n";
301     }
302     //////////////////////////////////double list////////////////////////////////////
303     struct d_node
304     {
305         int d data;
306         d_node *d next,*d last;
307     };
308     d_node *d np=NULL,*d head=NULL,*d tail=NULL;
309     void d_create_node()
310     {
311         cout<<"input new node:";
312         d np=new d_node;
313         cin>>d np->d data;
314         d np->d next=NULL;
315         d np->d last=NULL;
316     }
317     void d_create_list()
318     {
319         d np=new d_node;
320         cout<<"input data:";
321         cin>>val;
322         d np->d data=val;
323         d np->d next=NULL;
324         d np->d last=NULL;
325         if(d_head==NULL)
326             d_head=d_tail=d_np;
327         else
328         {
329             d_tail->d_next=d_np;
330             d np->d_last=d_tail;
331             d_tail=d_np;
332         }
333     }
334     void d_print_list()
335     {
336         d_node *d_temp = d_head;

```

```

337     while (d_temp != NULL)
338     {
339         cout << d_temp->d_data << " ";
340         d_temp = d_temp->d_next;
341     if (d_temp==d_head)
342         break;
343     }
344     cout << "\n\n";
345 }
346 void d_add_first()
347 {
348     d_create_node();
349
350     if (d_head == NULL) {
351         d_head = d_tail = d_np;
352     } else {
353         d_np->d_next = d_head;
354         d_head->d_last = d_np;
355         d_head = d_np;
356     }
357 }
358 void d_add_last()
359 {
360     d_create_node();
361     d_head->d_last->d_next=d_np;
362     d_np->d_last=d_head->d_last;
363     d_np->d_next=d_head;
364     d_head->d_last=d_np;
365 }
366 void d_del_first()
367 {
368     if (d_head == NULL) {
369         cout << "Empty list, nothing to delete.\n\n";
370         system("pause");
371
372         return;
373     }
374
375     d_node *d_c = d_head;
376
377     if (d_head == d_tail) {
378         d_head = d_tail = NULL;
379     } else {
380         d_head = d_head->d_next;
381         d_head->d_last = NULL;
382         d_tail->d_next = d_head;
383     }
384
385     delete d_c;
386 }
387 void d_del_last()
388 {
389     if (d_head == NULL)
390     {
391         cout << "Empty list, nothing to delete.\n\n";
392         system("pause");
393         return;
394     }
395
396     if (d_head == d_tail)
397     {
398         delete d_head;
399         d_head = d_tail = NULL;
400     }
401     else
402     {
403         d_node *temp = d_tail;
404         d_tail = d_tail->d_last;
405         d_tail->d_next = NULL;
406         delete temp;
407     }
408 }
409 int d_search_list()
410 {
411     int value,p=0;
412     cout<<"input search value:";
413     cin>>value;
414     d_tail=d_head;
415     while(d_tail !=NULL)
416     {
417         if(d_tail->d_data==value)
418         {
419             p=1;
420             break;

```

```

421     }
422     d_tail=d_tail->d_next;
423 }
424 return p;
425 }
426 void d_del_any()
427 {
428     if(d_search_list())
429     {
430         if(d_tail==d_head)
431         {
432             d_del_first();
433         }
434         else if(d_tail->d_next==NULL)
435         {
436             d_del_last();
437         }
438         else
439         {
440             d_tail->d_last->d_next=d_tail->d_next;
441             d_tail->d_next->d_last=d_tail->d_last;
442             delete d_tail;
443         }
444     }
445     else
446         cout<<"not found\n";
447 }
448 void d_add_after()
449 {
450     if(d_search_list())
451     {
452         if(d_tail->d_next==NULL)
453         {
454             d_add_last();
455         }
456         else
457         {
458             d_create_node();
459             d_np->d_next=d_tail->d_next;
460             d_tail->d_next->d_last=d_np;
461             d_tail->d_next=d_np;
462             d_np->d_last=d_tail;
463         }
464     }
465     else
466         cout<<"not found\n";
467 }
468 void d_add_before()
469 {
470     if(d_search_list())
471     {
472         if(d_tail==d_head)
473         {
474             d_add_first();
475         }
476         else
477         {
478             d_create_node();
479             d_tail->d_last->d_next=d_np;
480             d_np->d_last=d_tail->d_last;
481             d_np->d_next=d_tail;
482             d_tail->d_last=d_np;
483         }
484     }
485     else
486         cout<<"not found\n";
487 }
488 void d_update_node()
489 {
490     if(d_search_list())
491     {
492         int new_data;
493         cout<<"input new data:";
494         cin>>new_data;
495         d_tail->d_data=new_data;
496     }
497     else
498         cout<<"not found\n";
499 }
500 void d_sort_list()
501 {
502     d_tail=d_head;
503     while(d_tail !=NULL)
504     {

```

```

505     d_node *d_c=d_tail->d_next;
506     while(d_c !=NULL)
507     {
508         if(d_c->d_data < d_tail->d_data)
509             swap(d_c->d_data,d_tail->d_data);
510         d_c=d_c->d_next;
511     }
512     d_tail=d_tail->d_next;
513 }
514 }
515 void d_list_op_menu()
516 {
517     cout<<"    ---Double List---\n";
518     cout<<"Enter the transaction number\n\n";
519     cout<<" 1->Create list\n";
520     cout<<" 2->Add Last\n";
521     cout<<" 3->Add First\n";
522     cout<<" 4->Add After\n";
523     cout<<" 5->Add Before\n";
524     cout<<" 6->Del First\n";
525     cout<<" 7->Del Last\n";
526     cout<<" 8->Del Any\n";
527     cout<<" 9->Update Node\n";
528     cout<<" 10->Sort List\n\n";
529     cout<<"Press 0 to Main Menu\n\n";
530 }
531 //////////////////////////////////Main////////////////////////////////////
532 int main()
533 {
534     while(1)
535     {
536         system("cls");
537         main_maseg();
538         cin>>select;
539         system("cls");
540
541         if(select == 1)
542         {
543             stack_print();
544             stack_maseg();
545             cout<<"->";
546             cin>>op;
547             cout<<"\n";
548             system("cls");
549             while(1)
550             {
551                 switch(op)
552                 {
553                     case 1:
554                         cout<<"insert the value : ";
555                         cin>> val;
556                         stack_push(val);
557                         cout<<"\n";
558                         break;
559
560                     case 2:
561                         stack_pop();
562                         cout<<"\n";
563                         break;
564
565                     case 0:
566                         main();
567                         break;
568                 }
569                 system("cls");
570                 stack_print();
571                 stack_maseg();
572                 cout<<"->";
573                 cin>>op;
574                 cout<<"\n";
575             }
576
577             //////////////////////////////////
578
579             else if(select == 2)
580             {
581                 liner_q_print();
582                 liner_q_op_menu();
583                 cout<<"->";
584                 cin>>op;
585                 cout<<"\n";
586                 system("cls");
587                 while(1)
588                 {

```

```

589
590         switch(op)
591         {
592             case 1:
593                 cout<<"insert the value : ";
594                 cin>> val;
595                 liner_q_push(val);
596                 cout<<"\n";
597                 break;
598
599             case 2:
600                 liner_q_pop();
601                 cout<<"\n";
602                 break;
603             case 0:
604                 main();
605         }
606         system("cls");
607         liner_q_print();
608         liner_q_op_menu();
609         cout<<"->";
610         cin>>op;
611         cout<<"\n";
612     }
613 }
614
615
616 //////////////////////////////////////
617
618     else if(select == 3)
619     {
620         print_list();
621         linked_list_op_menu();
622         cout<<"->";
623         cin>>op;
624         cout<<"\n";
625         system("cls");
626         while(1)
627         {
628             switch(op)
629             {
630                 case 1:
631                     create_list();
632                     break;
633
634                 case 2:
635                     create_list();
636                     break;
637                 case 3:
638                     add_first();
639                     break;
640                 case 4:
641                     add_after();
642                     break;
643                 case 5:
644                     add_before();
645                     break;
646                 case 6:
647                     del_first();
648                     break;
649                 case 7:
650                     del_last();
651                     break;
652                 case 8:
653                     del_any();
654                     break;
655                 case 0:
656                     main();
657
658             }
659             system("cls");
660             print_list();
661             linked_list_op_menu();
662             cout<<"->";
663             cin>>op;
664             cout<<"\n";
665         }
666     }
667 }
668
669 //////////////////////////////////////
670
671     else if(select == 4)
672     {

```



```

673         d_print_list();
674         d_list_op_menu();
675         cout<<"->";
676         cin>>op;
677         cout<<"\n";
678         system("cls");
679         while(1)
680         {
681             switch(op)
682             {
683                 case 1:
684                     d_create_list();
685                     break;
686
687                 case 2:
688                     d_create_list();
689                     break;
690                 case 3:
691                     d_add_first();
692                     break;
693                 case 4:
694                     d_add_after();
695                     break;
696                 case 5:
697                     d_add_before();
698                     break;
699                 case 6:
700                     d_del_first();
701                     break;
702                 case 7:
703                     d_del_last();
704                     break;
705                 case 8:
706                     d_del_any();
707                     break;
708                 case 9:
709                     d_update_node();
710                     break;
711                 case 10:
712                     d_sort_list();
713                     break;
714                 case 0:
715                     main();
716
717             }
718             system("cls");
719             d_print_list();
720             d_list_op_menu();
721             cout<<"->";
722             cin>>op;
723             cout<<"\n";
724         }
725     }
726 }
727
728 }
729 return 0;
730 }
731

```