```
[*] update.cpp
1
      //updating na array
0
     #include<stdio.h>
     int main()
3
4 □
      {
 5
         int arr[30], element, num, i, location;
 6
 7
         printf("\nEnter no of elements :");
 8
         scanf("%d", &num);
         for (i = 0; i < num; i++) {
    scanf("%d", &arr[i]);</pre>
9 🖨
10
11
12
         printf("\nEnter the element to be inserted :");
13
         scanf("%d", &element);
14
         printf("\nEnter the location");
15
16
         scanf("%d", &location);
17
         //Create space at the specified location
         for (i = num; i >= location; i--) {
    arr[i] = arr[i - 1];
18 🖃
19
20
21
         num++;
22
         arr[location - 1] = element;
23
         //Print out the result of insertion
24
         for (i = 0; i < num; i++)
25
         printf("%d", arr[i]);
26
27
         return (0);
28 L }
```

```
//left shift na AARAY
#include<stdio.h>
int main()

# [5]
int A[5],i;
printf("ente 1)
         int A[5],i;
printf("ente the elemtas in an array");
7
          for(i=0;i<5;i++)
8
9 -
               scanf("%d",&A[i]);
12
          int temp=A[5];
          for(i=5;i<0;i--)
14 🖨
15
               A[i]=A[i-1];
A[0]=temp;
16
17
18
           for(i=0;i<5;i++)
19 🛱
20
21
22 - }
23 - }
           printf("%d",A[i]);
```

lift shifting array in dt.c

1

```
search elem. in list.cpp
 1
      //create display nad search element in linked list
 2
      #include<stdio.h>
 3
      #include<stdlib.h>
 4
      #include<conio.h>
 5
      struct node
 6 ₽ {
 7
          int data;
 8
          node *next;
 9 L };
10
     struct node *head=NULL;
11
     void create()
12 □ {
13
          char o;
14
     struct node * tail=NULL,*ptr;
15 🗀
              do{
16
              ptr=(node*)malloc(sizeof(node));
17
              if(tail==NULL)
18 🛱
19
                  head=ptr;tail=ptr;
20
21
              else
22 🗏
                  tail->next=ptr;
23
24
                  tail=ptr;
 25
                  tail->next=NULL;
26
27
              printf("enter no\n");
              scanf("%d",&ptr->data);
28
s 📶 Compile Log 🤣 Debug 🗓 Find Results
```

```
search elem. in list.cpp
             scanf("%d",&ptr->data);
28
29
             printf("press y for next node\n");
30
             o=getch();
31
32
             while(o=='y');
33
34
     void display()
35
36 ☐ {node * temp;
37
     temp=head;
         printf("the list is\n");
38
         while(temp!=NULL)
39
40 🛱 {
41
         printf("%d",temp->data);
42
         temp=temp->next;
43
44 [ }
45
46
     void search()
47 ☐ {node *temp;
     temp=head;
48
49
         int tem, count=0;
         printf("\nEnter the no. to find=");
50
         scanf("%d",&tem);
51
52
53
54
         while(temp!=NULL)
55 🛱
```

```
search elem. in list.cpp
          int tem,count=0;
printf("\nEnter the no. to find=");
scanf("%d",&tem);
49
50
51
52
53
54
          while(temp!=NULL)
55 🛱
56
          if(temp->data==tem)
57 🛱
58
              temp=temp->next;
59
          ++count;
60
              printf("\nelement found in node %d",count);
61
62
63
          else
64 🖨
              temp=temp->next;
65
              ++count;
              printf("\nelement not found in node %d",count);
66
67
   £,
68
69
70
71
     main()
72 □ {
73
          create();
74
          display();
75
          search();
76 L }
```

```
#include<stdio.h>
2
3
      int main()
4 □ {
           int A[5],i;
printf("enter aray");
for(i=0;i<5;i++)</pre>
5
6
7
8日
            {
9
                 scanf("%d",&A[i]);
10
            int temp=A[4];
for(i=4;i>=0;i--)
11
12 |
13 |
14
                 A[i]=A[i-1];
15
            A[0]=temp;
for(i=0;i<5;i++)
16
17
18 🖨
19 |
20 |
21 | }
                 printf("%d",A[i]);
            }
```

```
revrse an aaray.cpp
1
      //reverse an array
2
      #include<stdio.h>
3
     int main()
4 □ {
            int A[5],AR[5],ARR[5],i;
printf("enter the elements in 1 array");
for(i=0;i<5;i++)</pre>
5
6
7
8 🗗
9
                 scanf("%d",&A[i]);
10 -
11 |
12 |
            for(i=4;i>=0;i--)
13
14
15 }
                 printf("%d ",A[i]);
```

```
[*] queue1.cpp
 1
     //progamr for queue
 2
     #include<stdio.h>
 3
     #include<conio.h>
     #include<stdlib.h>
 5
     int q[5],front=-1,rare=-1;
     void enqueue()
 6
 7日 {
 8
         if(front==-1)
 9 🖨
10
             front++;rare++;
11
12
         else
13 🖨
             printf("enter value ");
14
             scanf("%d",&q[rare]);
rare++;
15
16
17
18
         if(rare>4)
19 🛱
         {printf("queue is full");
20
         return;
21
   L }
22
23
     void dequeue()
24 □ {
25
         if(rare==-1)
26
          printf("queue is empty");
27
          else
28
             printf("element removed is%d/n",q[front]);
```

```
27
             printf("element removed is%d/n",q[front]);
28
             front++;
29
30
                 }
31
32
33
     void view()
34 □ {
35
         int i;
36
         if(rare==-1)
         printf("queue is empty ");
37
38
         else
39日
40日
         for(i=front;i<rare;i++)</pre>
41
42
             printf("%d ",q[i]);
43
44 - }
    main()
46
47 □ {
         int no;
48
49
         while(1)
50 🖨
              printf("enter 1 to insert\nenter 2 to delete\nenter 3 to view \n");
51
              scanf("%d",&no);
52
              switch(no)
53
54日
s 🛍 Compile Log 🥏 Debug 🚨 Find Results
```

```
38
           else
          {
    for(i=front;i<rare;i++)
        printf("%d ",q[i]);
39日
41
42
43
44 - }
46
     main()
47 🖵 {
          int no;
while(1)
48
49
50 🛱
               printf("enter 1 to insert\nenter 2 to delete\nenter 3 to view \n"); scanf("%d",&no);
51
52
               switch(no)
53
54 🛱
55
                    case 1:enqueue();
                    break;
56
                    case 2: dequeue();
57
                    break;
58
                    case 3:view();
59
60
61
62
          }
63
64
65 L }
```

```
1 //DOUBLE LINKED LIOST
 2 #include<stdio.h>
 3 #include<stdlib.h>
 4 #include<conio.h>
 5
     struct node
 6 🖯 {
 7
         int data;
 8
         struct node*next,*prev;
 9 L };
10
    struct node *head=NULL;
11
     create()
12 [char opt; struct node *ptr, *temp;
13 T
         do
15
             ptr=(struct node*)malloc(sizeof(struct node));
16
             if(head==NULL)
17 🛱
                 ptr->next=NULL;
18
                 ptr->prev=NULL;
19
                 printf("enter the data");
20
21
                 scanf("%d",&ptr->data);
                 printf("press y to continue");
22
23
24
             else
25 🗀
                 temp=(struct node*)malloc(sizeof(struct node));
26
27
                 printf("enter the data");
                 scanf("%d",&temp->data);
28
20
s 📶 Compile Log 🤣 Debug 📮 Find Results
                      . .. ...
```

```
double linkes.cpp
24 |
25 |
             else
                  temp=(struct node*)malloc(sizeof(struct node));
26
27
                  printf("enter the data");
28
                  scanf("%d",&temp->data);
29
                  temp->next=NULL;
30
                  ptr->next=temp;
31
                  temp->prev=ptr;
32
                  ptr=temp;
33
                  printf("press y to cin");
34
                  opt=getchar();
35
36
37
         while(opt=='y');
38
   L }
     void insert()
39
40 □ {
41
          struct node *temp,*ptr;
42
          temp->next=ptr->next;
43
         ptr->next->prev=temp;
44
          ptr->next=temp;
45
          temp->prev=ptr;
46 L }
47
     int main()
48 🖵 {
49
          create();
50
          insert();
51 L }
```

```
[*] stack.cpp
      //STACK
 1
 2
     #include<stdio.h>
     #include<conio.h>
 4
     #include<stdlib.h>
      int top=-1,stack[5],i;
 6
     void push()
 7日 { if(top<4)
8日 {
 9
          top++;
          printf("enter value ");
10
          scanf("%d",&stack[top]);
11
12 | }
13 | }
     void pop()
14
15 🗗 {
16
17
          printf("element deleted is%d",stack[top]);
18
          top--;
19 L }
20
     void view()
21 🖯 {
22
          if(top==-1)
23
          printf("stack is empty");
24
          else
25
26 □ {
          printf("stack is ");
27
          for(i=0;i<=top;i++)
printf("%d" stock[i])</pre>
28
Compile Log 🖉 Debug 🗓 Find Results
```

```
[*] stack.cpp
23
          printf("stack is empty");
24
          else
25
26 🖯 {
         printf("stack is ");
for(i=0;i<=top;i++)</pre>
27
28
29
          printf("%d",stack[i]);
30
    - }
31
32 L }
33
     main()
34 ₽ (
          int n;
35
36
          while(1)
37 🖨
38
              printf("\nEnter 1 to push\nEnter 2 to pop\nEnter 3 to vieew\n");
              scanf("%d",&n);
39
40
              switch(n)
41 🗐
42
                   case 1:push();
43
                   break;
44
                   case 2:pop();
45
                   break;
46
                   case 3:view();
47
48
49
50 L }
```