ASSIGNMENT 6 (DS)

ASHI GOLYAN

1024240094

1.

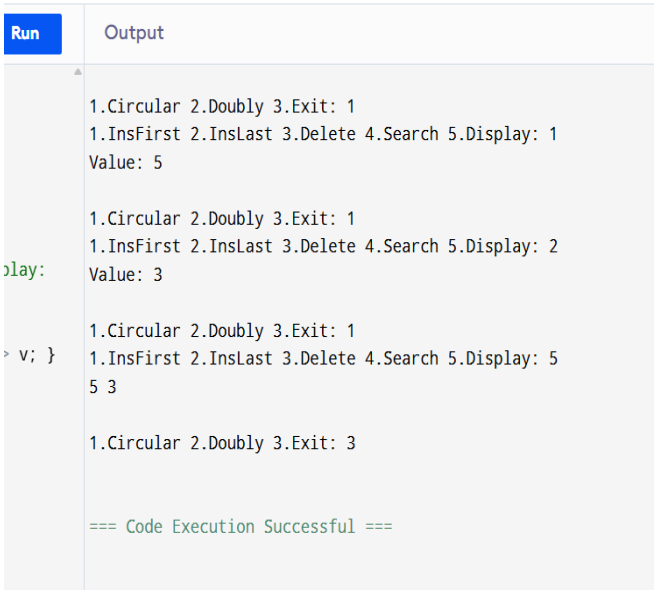
```
#include <iostream>
using namespace std;

struct NC { int data; NC* next; };
struct ND { int data; ND *next, *prev; };

NC* hc = NULL;
ND* hd = NULL;

void insFirstC(int v) {
   NC* n = new NC{v, hc};
   if (!hc) { hc = n; n->next = hc; return; }
   NC* t = hc;
   while (t->next != hc) t = t->next;
   t->next = n; n->next = hc; hc = n;
}

void insLastC(int v) {
   NC* n = new NC{v, NULL};
   if (!hc) { hc = n; n->next = hc; return; }
   NC* t = hc;
   while (t->next != hc) t = t->next;
   t->next = n; n->next = hc;
}
```

```
void delC(int v) {
   if (!hc) return;
   if (hc->data == v) {
      NC* t = hc;
      while (t->next != hc) t = t->next;
      if (hc == t) { delete hc; hc = NULL; return; }
      t->next = hc->next; delete hc; hc = t->next;
      return;
   }
   NC* t = hc;
   while (t->next != hc && t->next->data != v) t = t->next;
   if (t->next->data == v) { NC* d = t->next; t->next = d->next; delete d; }
}

void searchC(int v) {
   if (!hc) return;
   NC* t = hc; int p = 1;
   do {
      if (t->data == v) { cout << "Found at " << p << endl; return; }
      t = t->next; p++;
   } while (t != hc);
   cout << "Not found\n";
}

void dispC() {
   if (!hc) return;
   NC* t = hc;
   do { cout << t->data << " "; t = t->next; } while (t != hc);
   cout << endl;
}
```

```
void insFirstD(int v) {
   ND* n = new ND{v, hd, NULL};
   if (hd) hd->prev = n;
   hd = n;
}

void insLastD(int v) {
   ND* n = new ND{v, NULL, NULL};
   if (!hd) { hd = n; return; }
   ND* t = hd;
   while (t->next) t = t->next;
   t->next = n; n->prev = t;
}

void delD(int v) {
   ND* t = hd;
   while (t && t->data != v) t = t->next;
   if (!t) return;
   if (t->prev) t->prev->next = t->next; else hd = t->next;
   if (t->next) t->next->prev = t->prev;
   delete t;
}

void searchD(int v) {
   ND* t = hd; int p = 1;
   while (t) {
      if (t->data == v) { cout << "Found at " << p << endl; return; }
      t = t->next; p++;
   }
   cout << "Not found\n";
```

```
}

void dispD() {
   ND* t = hd;
   while (t) { cout << t->data << " "; t = t->next; }
   cout << endl;
}

int main() {
   int c, t, v;
   while (1) {
      cout << "\n1.Circular 2.Doubly 3.Exit: "; cin >> t;
      if (t == 3) break;
      cout << "1.InsFirst 2.InsLast 3.Delete 4.Search 5.Display: "; cin >> c;
      if (c <= 2) { cout << "Value: "; cin >> v; }
      else if (c == 3 || c == 4) { cout << "Value: "; cin >> v; }

      if (t == 1) {
         if (c == 1) insFirstC(v);
         else if (c == 2) insLastC(v);
         else if (c == 3) delC(v);
         else if (c == 4) searchC(v);
         else if (c == 5) dispC();
      } else {
         if (c == 1) insFirstD(v);
         else if (c == 2) insLastD(v);
         else if (c == 3) delD(v);
         else if (c == 4) searchD(v);
         else if (c == 5) dispD();
      }
   }
```

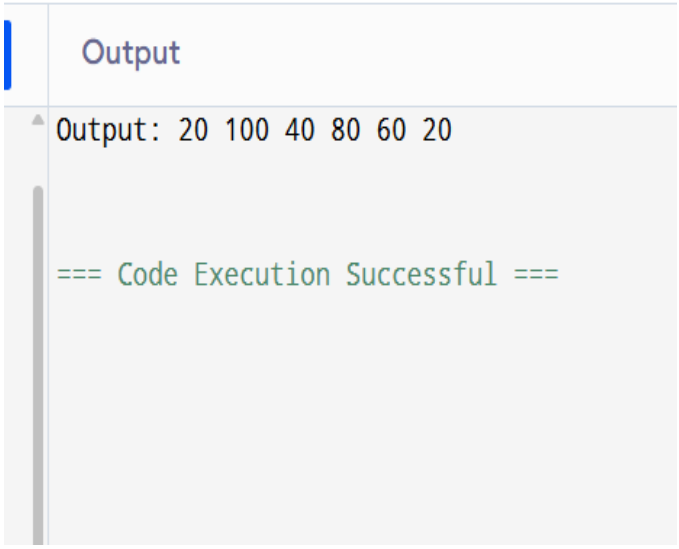
```
    return 0;
}
```



2.

```
#include <iostream>
using namespace std;

struct N { int data; N* next; };
N* h = NULL;

void ins(int v) {
    N* n = new N{v, NULL};
    if (!h) { h = n; n->next = h; return; }
    N* t = h;
    while (t->next != h) t = t->next;
    t->next = n; n->next = h;
}

void disp() {
    if (!h) return;
```

```
    N* t = h;
    do { cout << t->data << " "; t = t->next; } while (t != h);
    cout << h->data << endl;
}

int main() {
    ins(20); ins(100); ins(40); ins(80); ins(60);
    cout << "Output: ";
    disp();
    return 0;
}
```

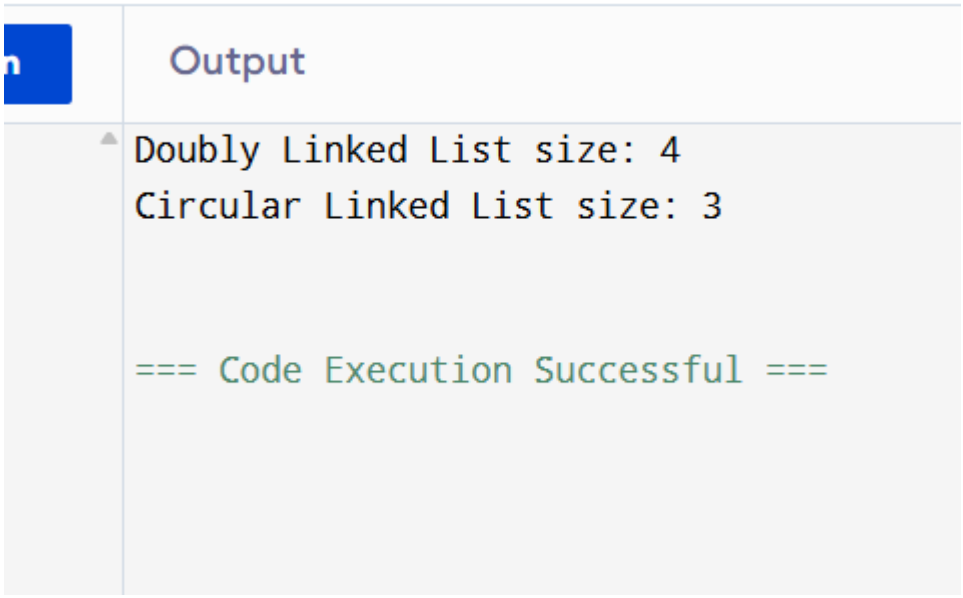


3.

```
#include <iostream>
using namespace std;

struct NC { int data; NC* next; };
struct ND { int data; ND *next, *prev; };

NC* hc = NULL;
ND* hd = NULL;
```

```
void insC(int v) {
   NC* n = new NC{v, NULL};
   if (!hc) { hc = n; n->next = hc; return; }
   NC* t = hc;
   while (t->next != hc) t = t->next;
   t->next = n; n->next = hc;
}

void insD(int v) {
   ND* n = new ND{v, NULL, NULL};
   if (!hd) { hd = n; return; }
   ND* t = hd;
   while (t->next) t = t->next;
   t->next = n; n->prev = t;
}

int sizeD() {
   int c = 0; ND* t = hd;
   while (t) { c++; t = t->next; }
   return c;
}

int sizeC() {
   if (!hc) return 0;
   int c = 0; NC* t = hc;
   do { c++; t = t->next; } while (t != hc);
   return c;
}

int main() {
```

```
    insD(10); insD(20); insD(30); insD(40);
    insC(5); insC(15); insC(25);

    cout << "Doubly Linked List size: " << sizeD() << endl;
    cout << "Circular Linked List size: " << sizeC() << endl;
    return 0;
}
```



4.

```
#include <iostream>
using namespace std;

struct N { char data; N *next, *prev; };
N* h = NULL;

void ins(char v) {
    N* n = new N{v, NULL, NULL};
    if (!h) { h = n; return; }
```

```
    N* t = h;
    while (t->next) t = t->next;
    t->next = n; n->prev = t;
}

bool isPal() {
    if (!h) return true;
    N *l = h, *r = h;
    while (r->next) r = r->next;
    while (l != r && l->prev != r) {
        if (l->data != r->data) return false;
        l = l->next; r = r->prev;
    }
    return true;
}

int main() {
    ins('L'); ins('E'); ins('V'); ins('E'); ins('L');
    cout << (isPal() ? "True" : "False") << endl;
    return 0;
}
```

5. #include <iostream>

```
using namespace std;

struct N { int data; N* next; };
N* h = NULL;

void ins(int v) {
   N* n = new N{v, NULL};
   if (!h) { h = n; return; }
   N* t = h;
   while (t->next) t = t->next;
   t->next = n;
}

void makeCirc() {
   if (!h) return;
   N* t = h;
   while (t->next) t = t->next;
   t->next = h;
}

bool isCirc() {
   if (!h) return false;
   N* t = h->next;
   while (t && t != h) t = t->next;
   return (t == h);
}

int main() {
   ins(2); ins(4); ins(6); ins(7); ins(5);
```

```
    makeCirc();
    cout << (isCirc() ? "True" : "False") << endl;
    return 0;
}
```