|  |  |
| --- | --- |
| **DANH SÁCH LIÊN KẾT ĐƠN** | **DANH SÁCH LIÊN KẾT KÉP** |
| 1. Khai báo cấu trúc dslk đơn List và Node với info thuộc kiểu số nguyên | |
| struct Node  {  int info;  Node\* next;  };  struct List  {  Node\* head;  Node\* tail;  }; |  |
| 2. Khởi tạo ds rỗng CreateList | |
| void CreateList(List& l)  {  l.head = NULL;  l.tail = NULL;  } |  |
| 3. Tạo 1 nút có info bằng X CreateNode (với X là 1 số nguyên được nhập từ bàn phím) | |
| Node\* CreateNode(int x)  {  Node\* p = new Node;  if (p == NULL) exit(1);  p->info = x;  p->next = NULL;  return p;  } |  |
| 4. Thêm 1 nút vào đầu dslk | |
| void AddHead(List& l, Node\* p)  {  if (l.head == NULL)  l.head = l.tail = p;  else {  p->next = l.head;  l.head = p;  }  } |  |
| 5. Thêm 1 nút vào cuối dslk | |
| void AddTail(List& l, Node\* p)  {  if (l.head == NULL)  l.head = l.tail = p;  else {  l.tail->next = p;  l.tail = p;  }  } |  |
| 6. Thêm 1 nút vào sau nút Q trong dslk | |
| void InsertAfterQ(list &l, Node \*p, Node \*q)  {  if(q!=NULL)  {  p->next=q->next;  q->next=p;  if(l.tail==q)  l.tail=p;  }  else  AddHead(l,p);  } |  |
| 7. Thêm 1 nút vào trước nút Q trong dslk | |
|  |  |
| 8. Hủy nút đầu trong dslk | |
| int RemoveHead(List& l, int& x)  {  if (l.head != NULL)  {  Node\* p = l.head;  x = p->info;  l.head = p->next;  delete p;  if (l.head == NULL)  l.tail = NULL;  return 1;  }  return 0;  } |  |
| 9. Hủy nút cuối trong dslk | |
|  |  |
| 10. Hủy nút sau nút Q trong dslk | |
| int RemoveAfterQ(List& l, Node\* q, int& x)  {  Node\* p;  if (q != NULL)  {  p = q->next;  if (p != NULL)  {  if (p == l.tail)  l.tail = q;  q->next = p->next;  x = p->info;  delete p;  }  return 1;  }  else  return 0;  } |  |
| 11. Hủy nút trước nút Q trong dslk | |
|  |  |
| 12. Duyệt dslk | |
|  |  |
| 13. Tìm kiếm | |
|  |  |