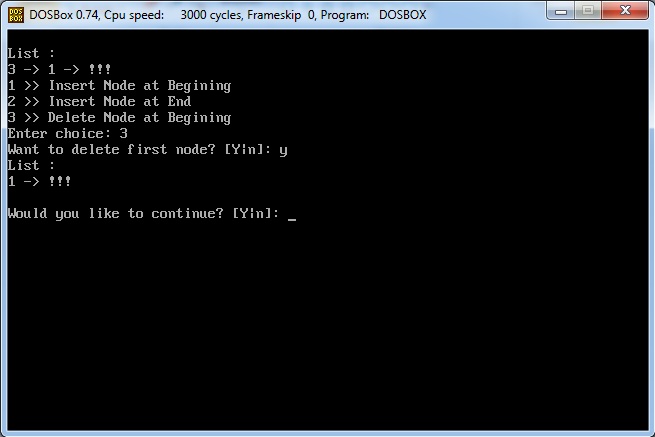
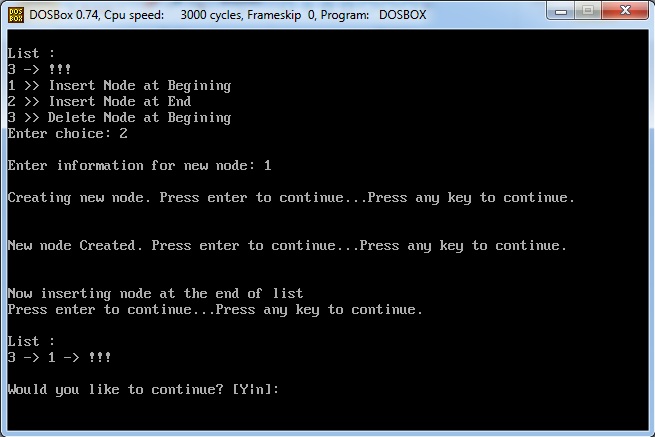
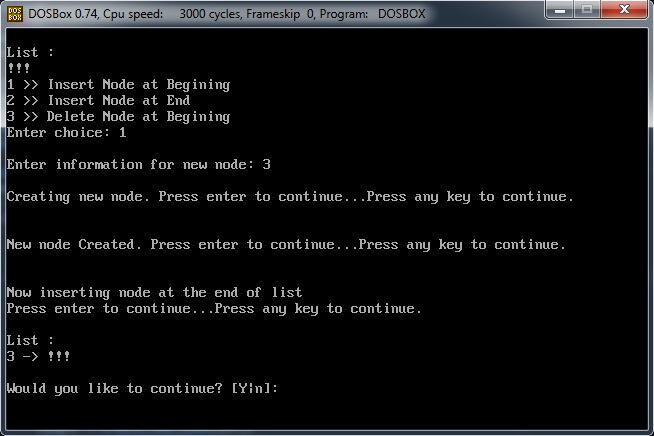
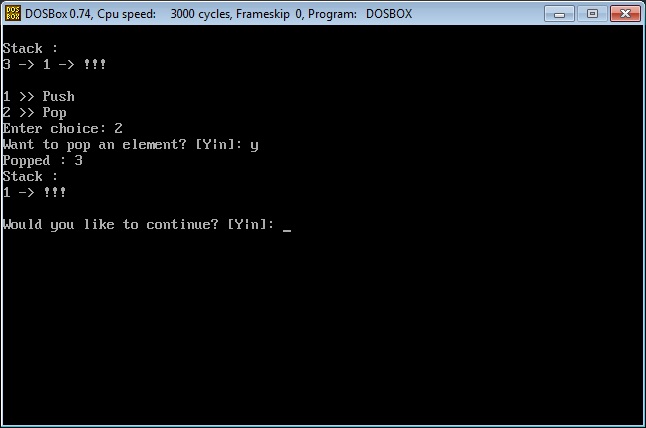
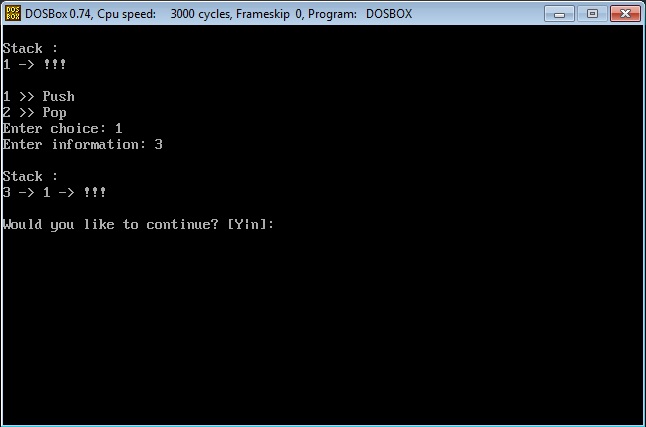
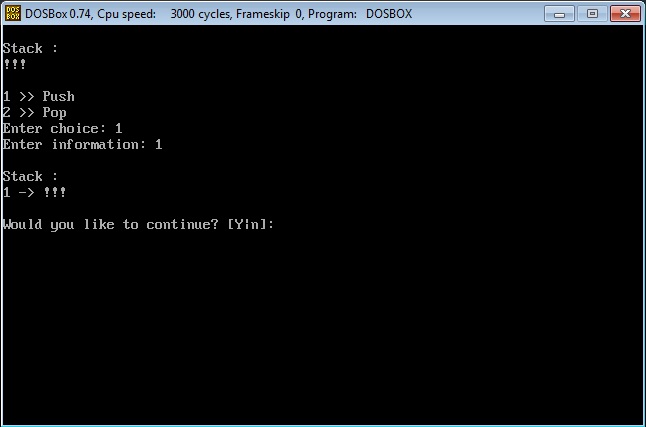
|  |
| --- |
| // Write a program to implement Insertion at begining, Insertion at end &  // Deletion at begining in a linked list containing an integer information  #include <iostream.h>  #include <stdlib.h>  #include <ctype.h>  #include <conio.h>  #include <process.h>  struct Node {  int info;  Node\* next;  } \*front, \*newptr, \*ptr, \*rear;  Node\* create\_new\_node(int);  void insert\_beg(Node\*);  void insert\_end(Node\*);  void display(Node\*);  void del\_node();  int main() {  front = rear = NULL;  int inf; char ch = 'y'; int c;  while (tolower(ch) == 'y') {  system("cls");  cout << "\nList :\n";  display(front);  cout << "1 >> Insert Node at Begining\n";  cout << "2 >> Insert Node at End\n";  cout << "3 >> Delete Node at Begining\n";  cout << "Enter choice: "; cin >> c;  if (c == 1 || c == 2) {  cout << "\nEnter information for new node: ";  cin >> inf;  cout << "\nCreating new node. Press enter to continue...";  system("pause");  newptr = create\_new\_node(inf);  if (newptr != NULL) {  cout << "\n\nNew node Created. Press enter to continue...";  system("pause");  } else {  cout << "\nCannot create new node. Aborting...\n";  system("pause"); exit(1);  }  cout << "\n\nNow inserting node at the end of list\n";  cout << "Press enter to continue...";  system("pause");  if (c == 1) insert\_beg(newptr);  else insert\_end(newptr);  cout << "\nList :\n";  display(front);  } else if (c == 3) {  int choice;  cout << "Want to delete first node? [Y|n]: "; choice = getche();  if (tolower(choice) == 'y') del\_node();  cout << "\nList :\n";  display(front);  } else  cout << "\nInvalid choice!\n";  cout << "\nWould you like to continue? [Y|n]: ";  ch = getche();  }  return 0;  }  Node\* create\_new\_node(int n) {  ptr = new Node;  ptr->info = n;  ptr->next = NULL;  return ptr;  }  void insert\_beg(Node\* np) {  if (front == NULL)  front = rear = np;  else {  np->next = front;  front = np;  }  }  void insert\_end(Node\* np) {  if (front == NULL)  front = rear = np;  else {  rear->next = np;  rear = np;  }  }  void display(Node\* np) {  while (np != NULL) {  cout << np->info << " -> ";  np = np->next;  } cout << "!!!\n";  }  void del\_node() {  if (front == NULL) cout << "\nUNDERFLOW!!!\n";  else {  ptr = front;  front = front->next;  delete ptr;  }  } |



|  |
| --- |
| // Write a program to implement operations on a Linked Stack  #include <iostream.h>  #include <stdlib.h>  #include <ctype.h>  #include <conio.h>  #include <process.h>  struct Node {  int info;  Node\* next;  } \*top, \*newptr, \*ptr;  Node\* create\_new\_node(int);  void push(Node\*);  int pop();  void display(Node\*);  int main() {  int inf; char ch = 'y';  top = NULL; int c;  while (tolower(ch) == 'y') {  system("cls");  cout << "\nStack :\n";  display(top);  cout << endl;  cout << "1 >> Push\n";  cout << "2 >> Pop\n";  cout << "Enter choice: "; cin >> c;  switch (c) {  case 1:  cout << "Enter information: "; cin >> inf;  newptr = create\_new\_node(inf);  if (newptr == NULL) {  cout << "Cannot create new node! Aborting...\n";  system("pause"); exit(1);  }  push(newptr);  cout << "\nStack :\n";  display(top); break;  case 2:  int choice;  cout << "Want to pop an element? [Y|n]: "; choice = getche();  if (tolower(choice) == 'y') {  int ele = pop();  if (ele != -1)  cout << "\nPopped : " << ele;  }  cout << "\nStack :\n";  display(top); break;  default:  cout << "\nInvalid Choice!\n";  }  cout << "\nWould you like to continue? [Y|n]: ";  ch = getche();  }  return 0;  }  Node\* create\_new\_node(int n) {  ptr = new Node;  ptr->info = n;  ptr->next = NULL;  return ptr;  }  void push(Node\* np) {  if (top == NULL) top = np;  else {  np->next = top;  top = np;  }  }  int pop() {  int inf = -1;  if (top == NULL) cout << "\nUNDERFLOW!!!\n";  else {  ptr = top;  top = top->next;  inf = ptr->info;  delete ptr;  } return inf;  }  void display(Node\* np) {  while (np != NULL) {  cout << np->info << " -> ";  np = np->next;  } cout << "!!!\n";  } |



|  |
| --- |
| // Write a program to implement operations on a Linked Queue  #include <iostream.h>  #include <stdlib.h>  #include <ctype.h>  #include <conio.h>  #include <process.h>  struct Node {  int info;  Node\* next;  } \*front, \*newptr, \*ptr, \*rear;  Node\* create\_new\_node(int);  void insert(Node\*);  void display(Node\*);  void del();  int main() {  front = rear = NULL;  int inf; char ch = 'y'; int c;  while (tolower(ch) == 'y') {  system("cls");  cout << "\nQueue :\n";  display(front);  cout << endl;  cout << "1 >> Insert\n";  cout << "2 >> Delete\n";  cout << "Enter choice: "; cin >> c;  switch (c) {  case 1:  cout << "\nEnter information for new node: ";  cin >> inf;  newptr = create\_new\_node(inf);  if (newptr == NULL) {  cout << "\nCannot create new node. Aborting...\n";  system("pause"); exit(1);  } insert(newptr);  cout << "\nQueue :\n";  display(front); break;  case 2:  int choice;  cout << "Want to delete node? [Y|n]: "; choice = getche();  if (tolower(choice) == 'y') del();  cout << "\nQueue :\n";  display(front); break;  default:  cout << "\nInvalid choice!\n";  }  cout << "\nWould you like to continue? [Y|n]: ";  ch = getche();  }  return 0;  }  Node\* create\_new\_node(int n) {  ptr = new Node;  ptr->info = n;  ptr->next = NULL;  return ptr;  }  void insert(Node\* np) {  if (front == NULL)  front = rear = np;  else {  rear->next = np;  rear = np;  }  }  void del() {  if (front == NULL) cout << "UNDERFLOW!!!\n";  else {  ptr = front;  front = ptr->next;  delete ptr;  }  }  void display(Node\* np) {  while (np != NULL) {  cout << np->info << " -> ";  np = np->next;  } cout << "!!!\n";  } |

