*#include*

*typedef struct node*

*{*

*int soilder;*

*struct node \* next;*

*}q;*

*q sol\_q;*

*q \*current,\*head,\*save;*

*int tot\_soilders=0;*

*void freelist()*

*{*

*int i;*

*current = head;*

*for (i=1;inext;*

*free(head);*

*head = current;*

*}*

*}*

*q \* getnode()*

*{*

*q \*temp;*

*temp = (q \*) malloc(sizeof(q));*

*if (temp == NULL)*

*{*

*printf("\n Memory allocation Failure");*

*exit(1);*

*}*

*else*

*return(temp);*

*}*

*void create\_list(int n)*

*{*

*q \*temp;*

*int i;*

*if (nsoilder = 1;*

*current->next = current;*

*head = current;*

*for (i=2;inext = temp;*

*current = temp;*

*current->soilder = i;*

*}*

*current->next = head;*

*tot\_soilders = n;*

*}*

*void display()*

*{*

*if (head == NULL)*

*{*

*printf("\nNo Soilders in the Queue");*

*return;*

*}*

*printf("%d",head->soilder);*

*printf("%c ",2);*

*current = head->next;*

*while( current != head)*

*{*

*printf("%d";,current->soilder);*

*printf("%c ",2);*

*current = current->next;*

*}*

*return;*

*}*

*q \*tail()*

*{*

*q \*temp;*

*current = head->next;*

*while (current != head)*

*{*

*temp = current;*

*current = current->next;*

*}*

*return(temp);*

*}*

*int left\_after\_sucide(int by\_n)*

*{*

*int i=1,j,dead\_sol;*

*current = head;*

*save = tail();*

*while (i<tot\_soilders)*

*{*

*for (j=1;jnext;*

*}*

*save->next = current->next;*

*if (current == head) head = current->next;*

*dead\_sol = current->soilder;*

*free(current);*

*display();*

*printf("\n\n%d%c is Dead \n%c RIP",dead\_sol,1,5);*

*getch();*

*current = save->next;*

*i++;*

*}*

*head = current;*

*display();*

*tot\_soilders = 1;*

*return(head->soilder);*

*}*

*main()*

*{*

*int ch,n;*

*head = NULL;*

*do*

*{*

*printf("\n1. For soilder list creation");*

*printf("\n2. For Displaying soilder list");*

*printf("\n3. For Sucide");*

*printf("\n0. For Exit");*

*scanf("%d",&ch);*

*switch(ch)*

*{*

*case 1:*

*printf("\nEnter the total no. of soilders");*

*scanf("%d",&n);*

*create\_list(n);*

*break;*

*case 2:*

*display();*

*getch();*

*break;*

*case 3:*

*if (tot\_soilders <= 1)*

*printf("There Should Be Atleast 2 Soilders in the List");*

*else*

*{*

*printf("\nEnter the no by which sucide is to be commited");*

*scanf("%d",&n);*

*if (n<1) printf("\nInvalid Number!");*

*else*

*printf("\nThe only Soilder left after "*

*"sucide session is %d%c",left\_after\_sucide(n),2);*

*}*

*getch();*

*break;*

*case 0:*

*return;*

*default :*

*printf("\nINVALID CHOICE");*

*getch();*

*}*

*} while (ch!=0);*

*freelist();*

*}*