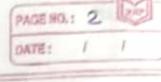
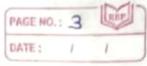


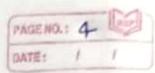
```
Programs
               Ac
1: Write a program to insert and delete an element at
  the nth and kth position in a linked list where or and
   Ic due taken from user.
  Hinclude <stdioon>
  # include <stalibit)
   Struct mode &
   int data;
   Struct mode* next;
   40
   Struct node + head;
   word insert (int data, int n) {
   Node*temp = new node();
   temp - data = data;
   temp -mext = NULL;
   if (n==1) }
   temp = next = heads
    head = temp;
    return;
   Void delete (int P);
   Struct mode * temp = head;
    if (P==1) }
   head = temp + next;
```



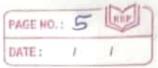
```
free (temp);
Node* temp = head;
for (int =0; i<n-2; i++) {
temp= temp = next;
temp - next = temp -next;
temp-next = temp;
void print();
for (int i=0; i<p-2; i++) {
 temp = temp = next;
 free (temp);
int main () }
int m, c, P;
head = NULL:
printf ("Enter the position of insertion);
Sconf ("(od", 4n);
Scanf ("%d", &c);
insest (c,n);
printf ("Enter the position of deletion);
scanf ("10d", 4P);
Delete (P);
```



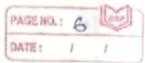
```
Print (c,0);
   retum:
2. Construct a new linked list by merging alternate nodes
  of two lists.
 Hinclude <stdiooh>
 #include < stallib h>
  Struct node
    int data;
    Struct data node * next;
  void printlist (struct mode * head)
     Struct node *pto = head;
     while (Pto)
  居
      printf (110/0d -> ", Pto-data);
      pto=pto-next;
     printf ("NULL\n");
  3
```



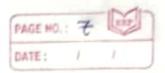
```
Void Push (struct moder+ head, int data)
  Struct mode * new mode = (struct node*) malloc
                       (size of(struct node));
  new node - data = data;
   new node - next = * head;
  *head = new node;
Struct node *shuffle merge (struct node *a, struct node *b)
  · Struct node dummy;
   Struct node + toil = 4 dummy;
   dummy next = NULL;
   while (1)
     if (a==NULL) {
       toil-next=b;
       break;
     else if (b==NULL) }
        tail-next=a;
        break;
     else ?
        toil + next = a;
```



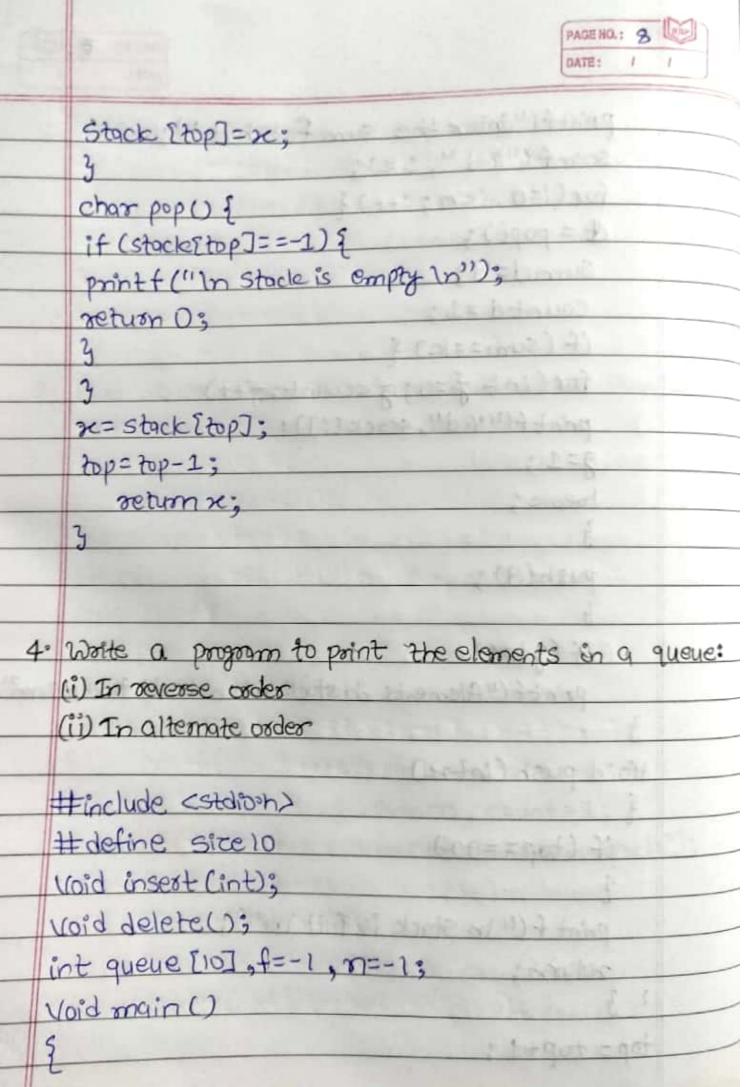
```
tail=a;
     a=a+next;
     tail - next = b:
     tail=b;
     b= b-mext:
   returndummy · next;
int main (void)
 int keys []= {1,2,3,4,5,6,733
 int n = size of keys / size of (keys [o]);
 Struct node *a=NULL, *b=NULL;
 for (int i=n-1; i>=0; (=i-2)
   push (4a, keys [:]);
 for (int i=n-2; i>=0; i=i-2)
   push (4b, keys [i]);
 printf ("first list");
 printlist (0);
 Printf ("Second List");
 printlist (b);
```

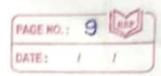


```
Struct node *head = Shufflemerge (a,b);
     printf("After merge");
     printlist (head);
     retumo;
3. Find all the elements in the stack whose sum is equal
  to k (where it is given from user).
  #include <stdiooh2
  int top=i-1;
  char stock [100];
  void push (intx);
  char pop();
  int main ()
   intain, b,d, k, g, sum=0, count=1;
   printf ("Enter the number of elements in Stack");
    Scanf ("%d", 4n);
    for (i=0; i<n; i++);
    printf ("Enter next element");
    Scanf (1006d", fb);
    push(b);
```



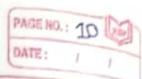
```
printf ("Giter the sum for cheek");
   scanf ("6d", fk);
   for (i=0; i<n; i++) {
   d = pop();
    Sumd=d;
    Countd = 1;
    if (sum == k) }
    for (int j=D; j < count; j++)
    printf("o(d", stack[i])3
    break;
    push(d);
   if (9!=1)
    printf ("Elements in stock will not add to the sum");
Void push (intre)
  if (top==99)
   print f ("In Stack is full In");
return;
top = top+1;
```



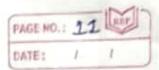


```
int value, choice;
  while (1) $
    printf (11/n/n **** MENU **** (n"))3
     printf (11 1. Insertion in 2. Deletion in 3. Print reverse)
          4. Print Alternate In 5. Exit");
    point f ("In Enter your choice");
     scanf ("10/0d", &choice);
     Switch (choice) 9
         Case 18
          printf ("Enter the value to insert");
          Scanf (" (od", 4 value);
          insert (value);
break;
          case 2:
          delete();
           break:
Cases:
           printf ("The Reversed queere is");
           for (int i=size; i>=0: i--)
         if (queuesij==0)
         continue;
         printf("%d", queuesiD;
         break;
```

ZI



	case 4:
	mulf ("nitemate elements of queue one").
	for (int i=0; icsize; i+=2)
catal	S. S. Control College and College and the State of the St
	if (queue [i]==0)
	continue;
	printf("%d", queuelis);
	3
	break;
	case 5:
	enit(0);
	default:
	printf ("In Wrong selection");
	У.
TOTAL	33
	Void insert Cint value) {
	if ((f==044 n==size-1) f==n+1)
10	printf ("In Queue is full and insertion can't be done
	else
	if(f==-1)
	f=0;
	0=(0141) 5% size;
	queue Con]=value;
	printf ("In Insertion Success");
	33
	Scanned with CamScanne



Void deleteus if (f==-1) Printf (" In Queue is Empty and Deletion cartbe done"); elses printf ("In Deleted 1/0d", queue[f]): f=(f+1) % size; if (f==n) f=m=-1; 33 5. (i) How among is different from linked list? The major difference between array and linked list regards to their structure. Arrays one index based bata structure where each element associated with an index. But in linked list, relies on refrences where each mode consists of the data and the refrences to the previous and next element.

(ii) write a program to add the first element of one list to another list.

include < stdiooh> # include < stdlib.h>

```
PAGE NO.: 12
Struct mode
   int data:
   struct node *next;
Void printlist (struct node * head)
   Struct node *ptr=head;
    while (Pto)
     printf (110/60), Ptr-data);
     ptr=ptr-next;
    printf ("Nallin");
4
Usid Push (struct node ** head, int data)
    struct node * new node = (struct node*) malloc
                  (sice of (struct node));
    new node - data = data;
    new mode + next = * head;
     *head = new node:
 4
Void move node (struct node ** dest ref, struct node **
                                 Sourceret)
```

```
PAGE HO.: 13
    if (*Source ref = = NULL)
      return:
    Struct node * new node = * source ref;
    *Source ret = (* source ref) -> mext;
     new node + next = *dest ref;
     * destref = newnode;
int main (void)
  int Keys [] = { 1,2,33;
  intn= size of (keys) | size of (keys [o]);
  Struct node * a= NULL:
   for (int i=n-1; i>=0; i--)
     push (fa, keys liD;
   Struct node * b = NULL;
   for (int i=0; icn; i++)
      push (4b, 2* keys (il);
   Move node (49,46):
   print ("first list");
    printlist (0);
    printf ("second list");
    printlist (b);
    zetum 0;
```