BSridhar AP19110010447 CSE-F.

Write a program to insert and delete an element at ith and kth Position in a linked list where n, k is taken from user. # include < stdio.h>

Stiller i ironink

V(1 + 1 Famer

Ell VIENV For

# include < std lib.h>

Struct Node <

int Statistics </br>

Struct Node \* next;
</br>

Struct Node \* head;

Void Insert (int statistics, int a) L.

Node # temp = new node ();

temp -> data = data;

if(a==1){

temp - next = head; ....

head = temp;

y Wetherny

Void Delete (int b) 2

Struct Node + temp = head;

if(b==1) <

head = temp - next;

free (temp);

return;

y

Node \* temp = head;

```
for (int 1=0; 1<0-2; 1++)
temp= temp - next;
     which is the first the first of the first of
 temp - next = temp -, next;
                           and the King Talk
  temp - next = temp;
                           1. July Luly (1)
  Void Print ();
                           - 1. HILAS HUMAN TE
 for (i=0; i<b-2; i++) {
                             JOHN BUILD
   temp = temp -> next;
                         free (temp);
                        · trans to the trans
4
 int main(){
                   that I was bout to wift
 int a, b, ke, this its that the trail the its book
  head = Null; I box won for to box
   Printf (" Enter the Position for inserting: ");
   Scanf ("Y. d", f n);
   Insert (x,a); then the hard with
    Printf(" Enter the Position forted elete: ");
    Scanf ("1.d", b);
             and Entry Black Box
   Delete (b);
              and - 100 & " Lack took P.
    Print(x);
    returno;
                 · box & fresh bush
                       March Control
```

```
construct a new linked list by merging the althoute nodes
of 21ists for ex: in list 1. 21213}, 11st 2. 24,516}.
    New list may be (1,4,2,5,13,69.
 # include estaio.h>
 # include a stdlib. h>
  Struct node &
    intdata;
    Struct node next;
    Void print list (struct node * head)
     Printfl"Y.d -,", (ptr -, data);
        Ptr=ptr-, next;
          Printf("Null/n"); }
     Voil Rush ( struct node * head, int data) Lov
      Strut node + new = (Struct node) malloc (Size of (Struct
       new - da = data;
                                                Mode);
       new - next= + head;
           * head = new;
      4
    Struct node * merge (struct node *a, struct node *b) &
        Strut riode * tail= wrong;
           wrong. next = Null;
          while (1) of
           if ( a = = Nul ) {
             tail - next = b;
            break;
```

```
else if (b= Null) {
                                                                                                                                                                  The Late Head by he dealers
                      tail -next=a;
                                                                                                                                                  THE BEST OF THE WAR TO BE THE STATE OF
                                                                                                                            Parante de la companya del companya della companya 
             break;
                  tail -next = a;
                                                                                                                                                                                                                       a trick of East of the
                                                                                                                                                                                                                      With a half the
                              break;
                                                                                                                                                                                                                                              Kirk takan Mi
                     elsel
                                           fail - next =a;
                                                                                                                                                                                                                    to be a book of
                                                tail = a;
                      tail → next-6;
                                               a=a -next;
                                                                                                            return wrong next;
                Void main () ? . book to be to
                         int Keys [] = {1,2,3,4,5,6,7};
                                  int x = Size of (Keys) / Size of Key [0];
                                  Struct node + a = Null; + b = Null;
                                  for (i= n-1; 170; i=1-a) {
                                                                  Push (da, keys(i));
                               for (int i=n-2; i=0; i=i-2)
                                                                      Push (4b, Iceys Ci));
                                Struct wode + head = Merse (aib);
                                                               Print list (head);
```

```
3. Find all the program of elements in the Stack whose Sumis
     Equel tok.
    # include c'stdio.h>
        int for =-1;
         int x;
        char stack [100];
         Void pash (int a);
         char pop ();
         int main () &
          int i, a, K, t, b, f, sum = 0, want = 1;
          Printf(" Enter the no. of dements in the stack!);
          Scant ("1.d", 4 x);
          for(i=0; ick; if +) ( , , )
              Printf(" Exter the elements to push in stack!);
               Scanf (" Y.d", da);
               Push (a);
         Printf ("Enter the Sum to be in the stack");
            Scanf (".r.d", db);
         for (1=0; 1< X; 1++) <
                               · Chalet of at
             t= Pop ();
              Sum += t;
              wunt +=1;
              1f (sum==b) L
              for ( j=0; j < 60 unt; j++) &
                 Printf("V.d", stack (i));
               4
```

```
Pugh (t) ?
            a differ the same for some relief to the fact
if (f(=1) {
  Printfl" The elements in the stack won't addup sum");
  void Push (inta) (
     if (top== 99) {
       Printfl" Stackis full (1/1);
     return;
    top=top+1;
   Stack [top] =oc;
    charpop ()
     if (stack (top))= xlicez
     char Pop ()
     if ( stack (top)) = =-1
  Printf (" Stack is Empty \n");
     y turno;
      N = Stack (top);
         tol= tol-1;
```

```
reverse order:
 Struct queue info !
    int tarray;
    int front;
    int rear;
     int space;
  5;
  det struct queue info tqueue.
    queue ( create queue (intmax))
     V = mallor ( size of (struit queue info));
     if ( Y= Null) {
      Printf(" Error");
     V- away = malloc (size of (int) max):
      if ( P - array == Null)
       Rintr(" Error");
      V-rarray = mallor
       V- Space = max-1;
       S→ froit =-1;
        V- rear = -1;
        return &;
       intis_full & ( Me ue a) X
         return ( ~ rear = ~ ~ space);
```

4)

```
int is_empty/queous/
  yetum (q- front = =-1);
void Enqueue (queueg, ind x) {
 frintf ("uhder-flow");
 retum;
 for (i= 9- front; ic=rear; i++) {
 Printf(" 1.d\f", g- array (i));
  intmain() L
  int max, x,i, choice, s, space, a, b, n=0;
  queque ;
   Plintf ("Enter the max" dements");
   Scant (" 1/d", &max);
    V = (realequeue (max);
    while (1) {
     Printf(" |n Menu! 1. Insert 2. Display reversed in order
                 3. Exist");
      Printf(" Ente the chore");
       Scanf ("1.d", & choice);
       Switch (choice) <
        casel:
       Printf ("Enter the dements!);
       Sconf (" y.d", & x);
        Enqueux ( s, x);
        break;
```

```
case 2:
                               To the day
Printf (" contents in the queve");
 display (x);
                                 K K L L
 for (i=o;ix Space;i+)2
                                 a = front and delete (v), s;
                                Push (a,s);
                  2 - fout =-1;
                   V-1 rear = -1;
                      1 the with
   for (i=0; iz space; i++)?
                      show of free with
     b = top and pop(s);
      Enqueue (2, b);
    Printfl'Revened
                 contentens are");
                    · (pro in the first
    display (4);
                    · (m. ) in the thirt
   exist(0);
                   · Comment of the Res
                   the format I don't
                    · (0 . p. - 1 + 1 . 1 . . )
                     · (LLJ)
```

```
ii) alternationer
   Hinclude < staro.h>
                               or of the ball to be
   Hindule estalib.h>
                                       : (y) factor
     Strut node L
                              the production of the second
       int data;
                               on the month of
       street node + next;
    Void Push (Struct node * head vef, charnew) &
      Struct node * node-new = (struct gode *) malloc(size of (
        node = new - data = new;
                                             struct nodes);
         node-new - next = (*head-ref);
         ( * head-ref) = node-new;
         Intmain () {
          Street node + head = NULL;
                                      him to the it
                Puph (Thead, 9);
                                       Push (dhead, 19);
                 Payh ( head, 29);
                 Puph ( thead, 39);
                 pushed head, 0);
                 Printfat(head);
                 returno;
        4
        void print alternate (struct rede * head &
                 jut count=0;
```

while (head! = NU(U) {

if (court 1/2=0) {

court chead -datace" ";

court ++;

head=head -hext;
}

y

- 5) i) Array consists of only single similar type of data, but the linked list consists of non-primitive data structure.
  - ii) Arrays belong, on the basis of indexes, and where by accessing the element must be in the format of declaring the location, but in the linked list, we can by the head and node for the structure.
  - iii) The accessing element in array is faster whereas in the linked list, it takes a bit slower.
  - iv) Arrays are in the fixed size, but inlinked lists, are dynamic and flexible, and increase its size.
  - v) The operations and like insertion, deletion in away, consume a lot of time. But in linked lists, the Performance of these like operations are fast enough.

A Land Land Comment of the

· E. L. A. Mary Bulle - The

Some Book

that track

```
5(Ti)
  #include cstatio.h>
 # include cstdlib.h>
                                  Roberton day to the
                                   F. . X
    Structhode (
                                 the week and book in
      int into;
      Struthode * next;
     Void Push ( struct Mode ** head ref, int new-info) 2
      Strut node * new-node = (strut node) *malloc (size of strut node)
       now-node-data = new-info;
    new-node-next=(* head-ref);

(*head-ref) = new-node;
                                           Property Sink
      Void printlist (struct node * head) {
        Struit node + temp = head;
         while (temp! = NULL) ?
           Print (". 1.d", temp data);
          temp=temp-next;
                                           Televisia EAR
                                     contra postr v
         Printf ("In");
                                    istoria a autor
           in the second of the
                                    1 converte to
                      tele 19
```