Program 1: Write a program to insert and delete an element at the new and kin position in a linked list where n and k wie taken from user. Explanation: A C program to insert and delete an element at user specified positions in a linked list. Algorithm: > Take input of the elements from the user! > Take input of the positions at which the number isto be inserted and deleted. > For insertion, create a new node whose next should point to the address of the next node. > For deletion, take a temp variable and shift the next of the desired node to the next node. -> Display the resultant linked list. or touth ) from the man to street full. Code: (111, d 9 1, 0 1) 11 # include < stdio.h > # include < stdlib.h> diene datal day in class mi struct node Acres 4-41) Tille men § int data; struct node \* next; 3\* new, \* head, \* tail, \* temp; void create\_list() int value, int ch; new = (struct node \*) malloc (size of (struct node)); printf ("Enter a number to form a list:"); Scanf (" 1.d", &value);

```
new > data = value;
          new > next = NULL)
          if ( head == NULL)
                            head = new;
                            tail = new;
                                                                    party of the or of the thing the
           else
                         tail > next = new;
                          tail = new;
           Printf ("Enter 1 to continue: ");
           scanf ("1.d", &ch);
    while (ch == 1);
                                                                      for keeps a comment in the first
                                                     el production de la contraction de la contractio
// Function to insert an element at a specified position
void insert (int in, int p)
                                                                               I shirt the law and firther I
               int i;
              new = (struct node *) malloc (size of (struct node));
             temp = head;
                                                                                                                     for (i=0; i2p-1; i++)
                      temp = temp > next;
          new -> data = in;
          new > next = temp > next;
         temp > next = new;
          printf ("\n ** INSERTION SUCCESS! ***\n");
                                                                                          mild to be deadle it!
 3
 // Function to delete an element at a specified position
 void delete (intp)
                     int i;
                     new = (struct node*) mallor (size of (struct node);
                      temp = head;
```

```
2 4.11
                for (i=o; i < p-1; itt)
                          temp = temp > next;
              temp > next = temp > next > next;
              Printf ("In * * * DELETION SUCCESS! * * * \m");
  3
1 Function to print the list
Void display()
                                                                    Traduction of the bearing and rates
 ξ
                                                                                        THE REPORT OF THE PARTY OF THE 
               temp = head;
               while (temp! = NULL)
                              Printf ("%d \n", temp → data);
                              temp = temp > next;
                 3
                                 and a grap oral part of the confidence of another
   y
   void main ()
                      int ins, pos, pos1
  S
                      create_list ();
                     printf ("Enter the number to be inserted:");
scanf ("1.d", &ins);
                      printf ("Enter the position:"); In I mi
                      scanf ("/d", & pos); Alguna a tal las all to to
                       insent (ins, pos); I diming aprolled 121 121 11
                       display (); the morning miderapping in the late of the
                       printf ("Enter the position at which number is to be deleted:");
                       scanf ("1.d", 60051); "1111 11 11 11 11 11
                       delete (pos 1); hamajanot in a lluma an allant
                        display ();
    Z
     output:
     Enter a number to form a list: 1
      Enter 1 to continue: 1
      Enter a number to form a list: 2
     Enter 1 to continue; 1
                        a number to forma list: 3
```

Enter 1 to continue; 2

```
Enter the number to be insented: 234
 Enter the position: 2
                                                                                        1:11 1117 1111
*** INSERTION SUCCESS ***
                                           The property of the property o
  2
 234
  3
Enter the position at which number is to be deleted: 3
* * * DELETION SUCCESS ***
 2
                                                                                                        (11)11 1 4010 1 21/11
 234
 Program 2:
                                                                                       The art - dering & della.
   Construct a new linked list by morging alternate nodes
   of two lists.
   Explanation:
   Create a new linked list by joining afternate nodes of
    two lists.
   Algorithm:
                                                              tron Sold Valence will relate at mil
-> Take a temp node as the start of the result list.
-> The tempnode gives tail something to point to initially
        when the gresult list is empty.
-> The pointer tail always points to the last node in the
        nesult list, therefore, appending new nodes!
-> The loop proceeds, removing one node from either a or b',
        and adding it two to tail.
-> Finally, the result is in temp. next.
 Code:
 # include < stdio.h >
# include < stdlib. h>
                                                           1 kd , and a manur
 struct nocle
                  int data;
                   struct node * neat;
 ¥;
```

```
1 A function to paint linked list
 S PrintList (struct node * head)
                       Struct node * ptr = head;
                       while (ptr)
                     & printf ("%d", ptr → data);
                                  ptr = ptr-> next;
                                                                                                                               to ignid and
                        printf ("NULL \n");
"Function to insert a new node to the linked list
Void insert (struct node ** head, int data)
ξ
                   struct node * newNode = (struct node *) malloc (sizeof (struct node));
                   newNode > data = data;
                   newNode > next = * head;
                   * head = newNode;
                                                                                                     the relation of the decision of the second o
                                                                                                               -11 1/2 2 11 111
 y
                                                                                                     the state of the state of the
1/ A function to merge alternate nodes of list using temp
void merge (struct node ** a, struct node ** b)
ટ્ડ
                 struct node temp;
               struct node * tail = & temp;
               temp. next = NULL; (11/1/ ) [11/1/ ) [1/1/1/ ]
               while (1)
                                                                                         if (*a = = NULL)
                           { tail -> next = NULL;
                                       break:
                          else if (*b==NULL)
                                       tail -> next = *a;
                                       break:
                         4
                         pl se
                                                             JIM .. H
                                    tail-next = *a;
```

```
tail = *a;
                      Add the middle not
    *a = (*a) \rightarrow next;
    tail -> next = * b;
                     and the state of
    tail = * b;
    *b = (*b) -> next;
  4
                        1000 2001 119
 મુ
 *a = temp. next;
                          " ( n/11711 ) 1/P+1
3
Struct node *a = NULL; * b = NULL;
   // construct first list
   for (int 1=3; i>=1; i--)
                       and the state of the state of
      insert (&a, i);
  " construct second list
                        North war of F 14
   for(inti=6; 1>=4; 1--)
      insert (&b,i):
  11 print both linked list
   printf ("first list");
   printList (a) july to the state of the bound of the first
   printf("Second list: ");
                          . garab - on Frank
   printList (b);
                1110/ 5 /11/1 1 1/20 1
   merge (&a, &b);
   printf ("In After merge: InIn").
   printf ("First list : ").
   printList (a);
                          (111)
   printf ("Second list: "); 11 11 11 11
   printList (b):
   return o;
output:
First list: 1 2
                  NULL
Second list: 4 5 6
                   NULL
After Merge:
 First list: 1 4 25 3 6 NULL
second list: NULL
```

Perogram 3: Find all the elements in the stack whose sum is equal to k. \$1111 TIN . A Explanation: Find all the possible set of elements in a stack whose sum and the desired is equal to k. Algorithm: > Take input of number of elements and each element value from the user. > Using nested loops, take the first element and store it in a variable ii, pop ii from the stack and store Store the remaining elements in stack 2. -> Store the first element of stack 2 in variable j. Now, add i and j to check if it is equal to k. -> If the sum is less than the expected value, we store it in a Stack "store" and continue the iteration until expected value and the sum are equal. -> Once the condition is satisfied, the elements are displayed. will be the strop of the all I as to be sell to test I all or Code: July 11 1 1 11 #include < stdio.h> ( 11 ) 9/11 #include< limits.h> et in rehimm the major of # define max 1000 A mark of A haline typedef struct stack (am 1 2) 21/24 int artmax1; int top; if William I. July a 101 it & think y 4 stack; void push (stack \*s, int data) neturn; s > an[s > top] = data; 4 I Agest 1884 il without

```
int pop(stack *s)
                 19 2 8 8 1 21 11 11 11 11 11 11
   if (5-> top < 0)
      return INT_MIN;
   S-> top -= 1;
   return s > ar[s > top+1];
4
Void display (stack * 5)
    for (i= s > top; i > -1; i--)
       printf ("xd", s > artij);
    Printf ("stack end \n");
           a of the soul by the second
4
int main (int auge, char const * argv[])
   Ş
   S1, top =-1;
   int expected, i, j, k, sum, n, num;
   pointf ("Enter the number of elements you want in the stack \n");
                                 Fil. 19
   scanf ("/d" 60);
   while (n--)
      printf ("number \n");
      scanf ("1,d", & num);
       push (& s1, num);
                                 1 1 1 1 1 1
   4
   printf ("Enter expected value \n");
   scanf ("1.d", & expected);
   while (ci = pop(&s1))! = INT_MIN)
        stack 52 = 51;
  £
        while ((j = pop(&52))! = INT_MIN)
             sum = i + j;
                      stack S3= Sa
            stack store;
            store. top = -1;
```

```
4.1
```

```
Push (&store, i);
       Push (&store, j);
       while ((s3.top!=-1) | (expected>=sum))
            if (sum = = expected)
                  display (bstore);
                  break;
            int temp;
           if ( (temp= pop(&53)) = = INT_MIN)
              break;
          Sum+ = temp;
          push (& store, temp);
       4
    4
 netwin 0;
4
Output:
Enter the number of elements you want in the stack: 4
number: 1
number: 1
number: 2
number: 3
Enter expected value: 5
23 stack end
1 1 3 stack end
```

```
Program 4.i.:
 Write a program to print the elements in a queue in
 heverse order.
                       at and your a motor of a thorage
 Algorithm:
Take input of all the elements of queue using enqueue
  function.
> In display function, do iteration from i=rear when
   1>= fromt.
> Display the queue.
 Code:
# include <stdio.h>
# include < stdlib.h>
# define SIZE 10
                                    A server training of the
int queque [SIZE];
int front, rear = -1;
                                     2 vylami sant
Void enqueue (int add)
                               . a million de la selat
ક
     if (hear == SIZE-1)
        Printf ("\n * * & QUEDE OVERFLOW * * * \n");
                                  - 901 clos 1 1 151
      plse
     € if (front ==-1)
           front=0;
         reartt:
        queue trear I = add;
             in the commentation will there is a state of a second second
3
void display()
     int i;
if (near == -1 11 front >rear)
کے
        printf ("In *** QUEVE IS EMPTY ** * \n");
     else
        for (i=rear; i>=front; i--)
           printf ("/d/t", queueli]);
```

```
rated one of
void main ()
  int add, ch;
      printf ("In Enter a number: ");
      scanf ("1.d", & add);
       enqueue (add);
       printf ("In Enter 1 to Continue:");
      scanf ("/d", &ch);
   while (ch == 1);
   Printf ("In The queue displayed in alternate order :\n");
   display ();
4
Output:
Enter a number : 1
Enter 1 to Continue: 1
Enter a number: 2
Enter 1 to continue! 1
Enter a number: 3
Enter 1 to continue: 2
The queue displayed in all heverse order:
 3 2 1
```

Porogram 4.11:

Write a program to print the elements in a queue in

Algonithm:

- -> Take input of all the elements of queue using enqueue function.
- In display function increment i' by 2 to print alternate elements.
- > Display the resultant queue.

```
Code:
# include <stdio.h>
# include < stdlib.h>
# define SIZE 10
int queue [SIXE];
int front, rear =-1;
Void enqueue (int add)
S
     if (rear == SIZE -1)
     PHINTH("\n *** QUEUE OVERFLOW *** \n");
     else.
       if (front = = -1)
          front =0;
        near ++;
                      of soil more than The 11
        queue trear ] = add;
     29
            1 11 (1 1/11)
Z
                               Legant deres a World of IN d
               Allinger old the
 void display () - I when I will
      int i; Realmani and the
S
                              Bern myrgin y fan till
      if ( hear == -1 | I fromt > rear )
        printf ("In QUEUE IS EMPTY In");
                                  else
       for (i=front; i = rear; i+=2)
           paintf ("/d/t", queueti]);
void main ()
                    the state of the part of the
     int add, ch;
     do
        printf ("In Enter a number: ").
        scanf ("1.d", & add);
        enqueue (add);
        printf (" In Enter 1 to continue : ").
        scanf ("1.d", &ch);
     while (ch == 1);
     printf ("In The queue displayed in alternate order: \n");
     display(); 2
```

Enter a number: 1
Enter 1 to continue: 1
Enter a number: 3
Enter 1 to continue: 1
Enter a number: 4
Enter a to continue: 2
The queue displayed in alternate order:

خ ۱

## Program 5.i:

How array is different from the linked list?

ARRAY

## A data structure consisting of a collection of elements each identified by array index.

- supports random access, so
  the programmer can directly
  access an element in the array
  using index.
- · Elements are stored in contiguous memory locations.
- · Programmer has to specify the size of the array at the time of declaring the array.
- Memory allocation happens at compile time; it is a static memory allocation.
- · Elements are independent of each other.

## LINKED LIST

A linear collection of data elements whose order is not given by their location in memory.

1 11 11 11 11

Supports sequential access, so the programmer has to sequentially go through each node until reaching the required element.

Elements can be stored anywhere in the memory.

There is no need for specifying the size of a linked list.

Memory allocation happens at nuntime; it is a dynamic memory allocation.

A node points to the next node or both next node and previous node

```
Broggiam 5.ii:
Write a program to add the first element of one list to
another list.
Algorithm:
                      the particularly of the
> Take two lists, for ext example, $1,2,34 and $4,5,64
-> To add the first node of one list to another, take
   the head pointer as return value and append it to
   the second list.
- To delete the used node from previous list, shift
   the head pointer to the next node.
-> Display the resultant list.
Code:
# include < stdio.h>
                                  and the drawn
# include < stallib.h>
struct node
S
  int data:
    struct node * next;
3;
struct node * new (int data)
    struct node * new_node = (struct node *) malloc(size of (struct node);
    new_node -> data = data;
     new_node → next = NULL;
    neturn new_node:
4
void push (struct node * * head, int new_data)
     struct node * new_node=(struct node *)malloc(sizeof(struct node));
     new_node >> next = (* head);
      (*head) = new-node;
```

```
" A function to add the first node to the other list
Void insert (struct node **first, struct node **second)
કૃ
  struct node * second_d=(struct node *) malloc (sixeof (struct node));
  second_d = * second;
  struct node * new= (struct node *) malloc (size of (struct node));
  new -> data = second_d -> data;
  new -> next = * first;
  *first = new;
  * second = second-d -> next;
20
void print (struct node *Node)
5
     while ( Node! = NULL)
            printf ("1.d", Node -> data);
            Node = Node - next:
                                    In It Burn I had
     Printf("\n");
2
int main ()
                             Krish diamental by the
    struct node * first = NULU;
    struct node * second = NULL;
   // create first list 1-2-3
      push (Infirst, 3);
      push (&first, 2);
      push (bfirst, 1);
     printf ("The first list is ");
     paint (first);
  11 create second list 4-5-56
    push ( b second, 6);
     push (b second, 5);
     push (&second, 4).
     printf ("Second list is").
```

```
Print (second);
    insert (&first, & second);
   Printf ("After adding an element from one list to another:
           In The first list is: ");
   Print (first);
   Printf ("The second list is: ");
   Print (second);
   return 0;
Output:
Figst list is 123
Second list is 45 6
After adding an element from one list to another:
The first list is: 4 1 2 3
The second list is: 56
```

4