DSA Assignment

Name : Balaji . G

Roll No : AP19110010517

Section : CSE-H

Write a program to insert and delete an element at the nth and kth position in a linked list where n and k is taken from the user.

```
# include <stdio.h>
# include < stdlib.h >
void ans (node*, int, int)
int size =0;
Struct node {
int data;
 struct node *next;
}
node * get node (int data)
{
   node *newnode = (struct node*) malloc(newnod
   new node → data=data;
   new node → next = null;
   return new node;
}
void ins (node* current, int pos, int data)
{
   if (Pos<1 || pos > size +1)
   printf ("Invalid");
   else
{  while (Pos--)
   {  if (pos==0)
      {
        node * temp = get node (data);
        temp→next = * Current;
        * Current = temp;
      }
```

```
else
 {
   current = &(*current)→next;
  }
 size++;
 }
}
void printf(struct node* head)
  {
    while (head! =null)
   {
     Printf("%d", head→data);
       head = head→next;
     }
      printf ("\n");
    }
 void del (struct node * head ~~def~~ , int pos) {
 if (head_ref == null)
 return;
  temp = head_ref;
 if (pos =0)
 {.
 * head_ref = temp→next;
 free(temp);
 return;
 for (int i=0; temp! =NULL && T<pos -1; i++)
   temp = temp→next;
   free (temp → next);
   temp → next = next ;
 }
  intmain ( )
  {..
```

```
Struct node * head = NULL;
push (&head, 7);
Push(& head, 8);
push ( & head, 6);
ins (&head, 7, 15);
del ( &head, 4);
Print list (head);
return (0);
}
```

Construct a new linked list by merging alternate nodes of two lists for example in list 1 we have {1,2,3} and in list 2 we have {4,5,6} in the new we should have {1,4,2,5,3,6

```
# include <stdio.h>
# ~~std~~include <stdlib.h>
Structnode {
   int data;
   Struct node* next;
 }
 Void printlist (structnode* head)
 {
   Structnode * ptr = head;
    while (ptr)
     {
      printf ("%d->", Ptr->data);
      ptr= ptr -> next; }
      Print f ("NULL\n");
    }
```

```
int keys [ ] = {1,2,3,4,5,6,7} ;
int n = size of (keys) / size of key [0];
stuct node* a = NULL , * b = NULL ;
for (int i = n-1 ; i>0; i = i-2)
    push (&a , keys(i]) ;
for ( int i = n-2; i >= 0 ; i = i-2)
    push ( &b , key[i]) ;
stuct node * head = merge (a,b);
printlist (head);
}
```

find all the elements in the stack whose sum is equal to K (where K is given by the user.

```
#include <stdio.h>
Void find ( int arr [ ] , int n , int s) {
  int sum = 0 ;
  int l=0, h = 0;
 for (l=0; l<n; l++) {
   while ( sum<s  &&h<n)
     Sum += arr (h];
      h++ ;
  if (sum == s)
  {  printf("found");
     return ; }
    Sum -= arr ( l];
 }
}
  int main (void) {
  int arr [ ] = {2,6,0,9,7,3}
   int s = 15;
  int n = size of (arr) / size of (arr[0]) ;
  find (arr, n, s) ;
  return 0;
```

Write a program to print the elements in a queue.

(i) In reverse order (ii) in alternate order

```
#include <stdio.h>
# include <stdlib.h>
struct node
{
  int data;
  struct node *next;
}
  void print rev(struct node *head)
{
  if ( head == NULL)
     return;
   Print rev (head-> next);
   Printf ("%d", head -> data);
Voidpush (struct node * headrev , char new)
{
 struct node *node_new = (struct node*) malloc
                                  (size of (struct node)
node_new -> data = new;
node_new-> next = (head_ref);
(*head_ref) = node_new;
}
 int main( )
   struct node *head = NULL;
    push (&head, 4);
    push (&head, 3);
    push (&head, 2);
   print rev(head); Print alternate (head);
   return 0;
 }
```

```
Void print alternate (Struct node* head)
{
  int Count =0;
   while (head l= NULL)
   {
      if (count % 2 ==0)
           Count << head → data << " ";
           Count ++;
           head = head → next;
     }
```

(i) How array is different from the linked list.

Answer:-

Key differences between Array and linked list

1) An array is a data structure that contains a collection of similar type data elements whereas the Linked list is Considered as non-Primitive datastructure Contains a Collection of Unordered linked elements Known as nodes.

2) (In the ~~(elements)~~ array the elements belong to indexes, i.e.., if you want to get into the fourth element you ~~wo~~have to write the Variable name with its index or location within the Square bracket.

3) (In a linked list through, you have to Start from the head and work your way through until you get to the fourth element.

4) Accessing an element in an array is fas while in linked list takes linear time, so it is ~~is~~ quite a bit Slower.

5) Operations like insertion and deletion in array Consume a lot of time. On the other hand the performance of these operations in linked list is fast.

6) (In a array, memory is assigned during
while in linked list it is

(ii)
```
#include <stdio.h>
#include <stdlib.h>
int len(int a[])
{ int i=0, an=0;
   while (1)
   {
     if(a[i])
     {
       an++, i++;
     }
     else
      {
        break;
       }
     }
     return an;
   }
    Void changinglist (int a[a], int b[])
   {
     for (int i=len(a)-1; i>=0; i--)
    {
       a[i+1] = a(i);
    }
```

Scanned with CamScanner

```
a[0] = b[0];
Printf ("\n the elements of first array :\n",
for (int i=0 ; i< len(a); i++)
 {
    Printf ("%d", a[i]);
 }
 for (int i=0 ; i< len(b); i++)
  {  b[i] = b[i+1] ; }
  Printf ("\n the elements of second array:\n",
  for (int i=0 ; i<len(b) ; i++)
 { Printf ("%d", b[i]);

  }
}
 int main()
  {
    int a[10] = {1,2,3} ,b[10] = {4,5,6} ;
   chaugınglist (a,b);
}
```