Y. Divya Sri
AP19110010036
CSE-G

Assignment-4

① 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 user.

Ans:
```
# include <stdio.h>
# include <stdio.h>
Struct node {
     int data;
     Struct node* next;
};
int main()
{
     int i, n, item, pos, dpos;
     Struct node *a, *b, * head, *newnode ,* nextnods;
     printf ("Enter no. of nodes you want: ");
     Scanf ("%d", &n);
     printf ("Enter the value for the head node: ");
     scanf (" %d" , &item);
     int count = 1;
     b = (Struct node*) mallac (size of (struct node));
     b → data = item;
     b → next = NULL;
     head = b;
      a = head;
     for (int i = 1; i<n; i ++)
     {
      printf (" enter the value for the next node: ");
      scanf ("%d", & item);
      b = (struct node *) malloc (size of (struct node));
      b→data = item;
      b → next = NULL;
      count ++;
      a→ next = b;
      a = a → next;
      }
```

```
        a = head;
        while ( a!= NULL)
        {
            printf(" the elements before inseeting an ele :-%d\n", a->data);
             a= a-> next;
        }
        printf ("entee the position you want to entee elements: ");
        scanf( "%d", &pos);
         a= head;
         if (d pos > count)
         {
            printf("Invalid\n");
          }
          else {
          int  k =1;
          a = head;
          while (k<=pos)
          {
             next node  = a;
               a =a-> next;
               k ++;
          }
          nextnode -> next = a -> next;
          printf ("deleted  element %d\n", a->data);
          free (a);
         }
          a= head;
         while (a!= NULL)
        {
           printf(" the elements after deleting an element:%d\n", a->data);
            a = a-> next;
        }
    }
```

② Contruct a new linked list by merging alternate nodes of two lists fol example in list1 we have {1,2,3} and in list 2 we have {4,5,6} in the new list we should have {1,4,2,5,3,6}.

Ans:-

```
# include <stdio.h>
# include <stdio.h>
 struct node {
       int data;
       struct node*next;
 };
 int  main()
 {
       int i, n, item1; item2, m;
       struct node  *p, *q, *head1, *head2, *r, *s, *temp, *a1, *b1;
       printf("enter no of nodes you want for first list : ");
       scanf("%d", &n);
       printf("enter the value for the head node: ");
       scanf("%d", &item1);
       q = (struct nod*)malloc(size of (struct node));
       q->data = item1;
       q->next = NULL;
       head1 = q;
        p = head1;
       for(int i=1; i<n; i++)
       {
       printf("enter the value fol the next node: ");
       scanf ("%d",&item1);
       q= (struct node*) malloc(size of (struct node));
       q->data = item1;
       q->next = NULL;
       p->next = q;
       p = p->next;
       }
       p = head1;
printf("enter no.of nodes you want for second list :");
scanf("%d", &m);
printf("enter ~~no.o~~ the value for the head node : ");
scanf ("%d" , &item 2);
s = (struct node*) malloc ( size of (struct node));
s -> data = item2
s -> next = NULL;
head2 = s;
 r = head2;
for ( int i = 1; i<m; i++)
{
  printf ("enter the value for the  next node");
  scanf (" %d", &item2);
  s = (struct node*) malloc (size of ( struct node));
  s -> data = item2;
  s -> next = NULL;
  r -> next = s;
  r = r -> next;
 }
 r = head2
 temp = p;
 while ( p != NULL && q != NULL)
 {
   a1 = p -> next;
   b1 = r -> next;
   p -> next = r;
   r -> next = a1;
   p = a1;
   r = b1;
 }
 while (temp != NULL)
 {
   printf("The alternate merged linked list values = %d\n", temp
                                                           -> data);
   temp = temp -> next;
 }
}
```

3) Find all the element in the stack whose sum is equal to k. (where k is given from user).

Ans:-

```
 # include <stdio.h>
  int  main()
 {
    int  n, stack[100], top = -1, x,k,sum,y;
   .printf (" enter no. of elements you want to enter in to stack:");
    scanf ("%d", &n);
    for (int i=0; i<n; i++)
    {
       if ( top >= n-1)
    {
       printf("\n\t stack is over flow ");
    }
    else
    {
        printf (" enter a value to be pushed: ");
        scanf ("%d", &x);
        top ++;
        stack[top] = x;
     }
    }
     printf (" enter the sum you want for:");
     scanf("%d", &k);
     if ( top<= -1)
    {
        printf ("\n\t stack is under flow");
    }
    else
    {
        while (top!=-1)
        {
          y = stack[top];
          sum =0;
          while (y!=0)
          {
            sum = sum + (y%10);
            y = y/10;
          }
        if (sum==k)
       {
           printf("The element is =%d \n",stack [top]);
       }
       else
       {
         printf("element not found ");
       }
       top--;
       }
    }
 }
```

4 Write a program to print the elements in a queue.

1. In reverse order

2. in alternate order

Ans:

```
  # include <stdio.h>
  # define MAX 100
   int queue_array[MAX];
   int stack[100];
   int top=-1;
   int rear = -1;
   int front = -1;
   int main()
   {
      int n;
      printf("Enter no.of elements you want to enter into the queue");
      scanf("%d", & n);
      for (int i=0;i<n;i++)
      {
        int add_item;
        if (rear ==MAX-1)
           printf("Queue overflow \n");
        else
        {
           if (front ==-1)
            front =0;
      printf(" Insert the element in queue :");
      scanf (" %d", &add_item);
      rear = rear + 1;
      queue_array [rear] = add_item;
    }
  }
   int choice;
   printf("1. To print elements in a reverse order\n
           2. To print elements in a alternate order.\n");
   printf(" Enter your choice");
   scanf("%d",&choice);
   switch (choice)
  {
    case 1:
    {
      for (int i=0; i<n; i++)
      {
      if (front == -1 || front >rear)
        {
          printf(" queue underflow\n");
        }
        else
         {
           top ++;
           stack [top] = queue_array [front];
           front = front + 1;
           }
        }
         if (top>=0)
          {
            printf("\n.The elements in queue in reverse order\n");
            for(int i=top; i>=0; i--)
            printf("\n%d", stack[i]);
            printf("\n");
          }
          else
          {
```

```
          printf ("\n the stack is empty");
          }
          break;
        }
     case 2:
     {
        for (int i = front ; i<= rear ; i+ =2)
        printf ("%d\n", queue_array [i]);
        break;
     }
     default:
     printf(" entel a valid choice");
     break;
   }
}
```

5.) i.) How ~~many~~ array is different from the linked list.

Ans: ~~Ar~~ The key difference between Array and linked list is,

1. An array is a data structure that contains a collection of similal ~~data~~ data type data elements whereas the linked list is considered as non-premitive data structure contains a collection of unordered linked elements known as nodes.

2. In the array the elements belongs to indexes. i.e, if you awant to get into the fourth element you 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 head and work your way through until you get to the fourth element

4. ~~Accolding~~ Accessing an element in an array, is fast, while in linked list takes linear time, so it 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 compile time while in linked list it is allocated during execution of run time.

ii) write a program to add the first element of one list to another element list for example we have {1,2,3} in list 1 and {4,5,6} in list 2 we have to get {4,1,2,3} as output for list1 and {5,6} for list 2.

Ans:-

```
# include <stdio.h>
# include <stdlib.h>
int len ( int a[ ])
{
  int i = 0, a, ~~m=0~~; an=0,
  while (1)
  {
   if (a [i])
   {
     an ++, i++;
   }
   else
    {
     break;
    }
  }
  return an;
}
void changing list ( int a[ ] ,int b[ ]).
{
 for ( int i = len(a)-1; i>=0; i--)
 {
  a[i+1] = a[i];
 }
 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] = [11,2,33}, b[10] = {14,57,6}
   Changing list (a,b);
  }
```