ASSIGNMENT - 4

P. Ruthik Jenkat
AP19110010549

Problem:- 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.

Explanation of problem: The aim of this program is to create a linked list and the user must give the position where to insert and delete then the program must display the linked list after insertion and deletion is done.

Algorithm:-

* Start
* Create a linked list using structures
* Give the condition to enter the no. of variables used
* And then enter the values and Enter the positions where to Insert and delete
* Give the condition to print after the insertion and deletion is done
* Stop

Program in C:-

```
#include<stdio.h>
#include<stdlib.h>
Struct node
 {
Struct node *next;
 };
Struct node *curr, *temp;
Void inspos(struct node*);
Void delpos(struct node*);
Void main(void)
 {
```

```
struct node *s;
int ch;
S=NULL;
do
 {
  printf ("\n 1.Inspos\n");
  printf (" 2.Delpos\n");
  printf (" 3.Exit\n");
  printf (" Enter the choice");
  scanf (" %d", &ch);
     switch (ch)
        {
        case 1 : inpos(s);
             break;
        case 2 : delpos(s);
             break;
        } which (ch != 3)
 }
    Void inpos (struct node *x)
     {
     int pos, c=1;
     curr = x;
     printf("Enter the value to be inseited: ");
     Scanf (" %d", &pos);
       while (curr→next! =NULL)
          {
          c++;
          if (c==pos)
          {
          temp = (struct node *) malloc (sizeof(struct node
          printf ("Enter the number:");
          Scanf("%d", &temp→n);
          temp→next = curr→next;
          curr→next = temp;
```

```
            break;
            }
        }
    }
Void delpos (struct node * x)
 {
 int pos, c=1;
 curr = x;
printf ("Enter the position to be deleted: ");
Scanf ("%d", &pos);
 while (curr→next != NULL)
    {
    c++;
     if (c==pos)
       {
      temp = curr→next;
      curr→next = curr→next→next;
       free(temp);
       }
      curr = curr→next;
       }
    }
```

Sample input and Output:

Enter the No. of Variables to be used - 6
Enter the values - 1
2
3
4
5
6
Enter the position to be inserted - 3
Enter the number to insert - 8
Enter the position to be deleted - 5

Output

The list after deletion and inseation is 1 2 8 3 5 6

Explanation of output:

The o/p of this program is pointed by satisfying all the conditions entered in the program.

Program: Construct a new linked list by merging alternate nodes or two lists for example in list 1 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}

Explanation of the problem: The main theme of this program is to create two linked lists and merge them The input will be given as Enter the elements in list 1 and list 2 and then it merges. For merging the two linked lists we have to write the condition.

* Start
* Create two linked lists using structure
* Write the condition to enter the numbers in the list
* And then write the condition to merge the lists
* Print the output
* Stop.

```
#include <stdio.h>
#include <stdlib.h>
Struct node {
        int data;
        Struct node *link;
 }*head1 = NULL, *temp, *temp1, *head2 = NULL, *head3 = NULL;
Struct node *insert (Struct node *head, int x)
 {
 temp = (Struct node *) malloc (size of (Struct node));
 temp → data = x;
 temp → link = NULL.
 if (head = = NULL)
    {
     head = temp;
    }
```

```
    else
     {
     temp1 = head
     while (temp1 → link != NULL)
       {
        temp1 = temp1 → link;
       }
      temp1 → link = temp;
       }
       return head;
   }
  int main()
   {
  int p, q, x, i;
printf("Enter the No. of elements of linked list1 ");
scanf("%d", &p);
for(i=0; i<p; i++)
  {
printf("Enter the element");
scanf("%d", &x);
head1 = insert(head1, x);
}
printf("Enter no. of elements of linked list2 ");
scanf("%d", &q);
for(i=0; i<q; i++)
  {
  printf("Enter the element");
  scanf("%d", &x);
  head2 = insert(head2, x);
 }
 temp1 = head1;
 head1 = head2; temp2 = head2
 while(temp1!=NULL && temp2 != NULL)
```

```
{
printf("%d", temp1->data);
printf("%d", temp2 -> data);
temp1 = temp1 -> link;
temp2 = temp2 -> link;
}
while (temp1 != NULL)
{
printf ("%d", temp1->data);
temp1 = temp1 -> link;
}
while (temp2 != NULL)
{
printf ("%d", temp2 -> data);
temp2 = temp2 -> link;
}
```

```
enter the No.of elements in list1 = 3
1
2
3
enter the no.of elements in list2 = 3
4
5
6
```

new list
{1,4,2,5,3,6

The output is printed by entering all the values and satisfying all the conditions the links are merged.

3. Problem: Find all the elements in the stack whose sum is equal to k where k is given from user.

Explanation of problem: The Aim of this problem is to create a linked list and enter the elements in the stack and enter a value from user then it prints sum of values which is equal to value entered by user

Algorithm

* start
* Enter the No. of variables used in stack and Enter values
* Enter the values from user to check the sum
* write the conditions in the stack to print the number of sum which is equal to entered by user.
* print the numbers.
* stop.

program in c:

```
#include <stdio.h>
int s1[10], top1 = -1, s2[10], top2 = -1;
int s1 empty()
  {
    if (top1 == -1)
        return 1;
    else
      return 0;
   }
int stop()
  {
    return s1[top1];
  }
int s1push(int x)
   {
```

```
    s1[++top1]=x;
  }
int s1empty()
   {
   if(top2==-1)
     return 1;
   else
     return 0;
  }
int s1top()
    {
    top2--;
    }
int s2push(int x)
   {
  s2[++top2]=x;
  }
int sum(int k)
   {
   int x;
  while(s1empty()!=1)
    {
    x=s1top();
    s1pop();
   while(s1empty()!=1)
    {
    if(x+s1top()==k)
     {
    printf("%d,%d",x,s1top());
     }
   s2push(s1top());
   s1pop();
  }
```

```
    }
    while (s2.empty() != 1)
       {
       s1.push(s2.top());
       s2.pop();
       }
     }
   }
int main()
  {
  int n,i,e,k;
  printf("enter the no.of elements of stack : ");
  scanf("%d",&n);
  for(i=0;i<n;i++)
   {
   scanf("%d",&e);
    s1.push(e);
   }
  printf("enter the values of constant sum : \n");
  scanf("%d",&k);
  printf("The combinations whose sum is equal to k i.e");
   sum(k);
  }
```

Sample Input:

Enter the no.of elements of stack - 5

Enter the values - 1
2
3
4
5

Enter the value of constant sum k - 5

Sample Output :

The combination whose sum equal to k is

(1, 4)

(2, 3)

The output of this program is printed by pushing the elements into the stack and By entering the value the sum of values are printed.

4) Problem: Write a program to print a elements in a Queue
i, reverse order
ii, in alternate order

i, Explanation of problem: The Aim of this program is to print the elements in the Queue in reverse order.

Algorithm:
* Start
* Create a Queue and enter the no. of variables and values
* Write the condition to push the values
* Write the condition to reverse the elements in the Queue
* Stop

Program in c:

```
#include <stdio.h>
int p[20], front = -1, rear = -1;
int s[10], top = -1
int p push(int x)
    {
    if (front = = -1)
    {
     p[++rear] = x;
      front++;
    }
   else
     p[++rear] = x;
   }
 int p front()
     {
```

```
    return p[front];
}
int ppop()
    {
    front++;
    }
int pempty()
     {
     if (front>rear)
        return 1;
      else
        return 0;
}
int spop()
    {
     top--;
    }
int spush(int x)
     {
     s[++top]=x;
    }
int sempty()
    {
  if (top==-1)
      return 1;
     else
      return 0;
}
int stop()
     {
     return s[top];
    }
int main()
    {
```

```
Int n,i,x;
 printf(" enter the no.of elements of Queue ");
 Scanf("%d", &n)
 for(i=0; i<n; i++)
  {
  scanf("%d", &x);
   ppush(x);
  }
 while(pempty()!=1){
   Spush(pfront());
    ppop();
   }
 while(sempty()!=1)
   {
  printf("%d", stop());
   spop();
   }
}
```

Sample input:

enter the elements of Queue 5

enter the values to EnQueue - 1
2
3
4
5

Sample Output

In reverse Order - 5
4
3
2
1

Explanation of output: The output is printed by entering the elements in 'p' and copying the elements to 's' and then reversed by satisfying all the conditions.

ii, Alternate order:

Explanation of problem: The Aim of the program is to enQueue the elements in Queue and print the alternate elements from Queue

*Start
* Create a Queue using array and enter the no.of Variables
* EnQueue the values to the Queue
* Write the condition to print the alternate elements in Queue
* Stop

```
#include <stdio.h>
int p[20], front = -1, rear = -1;
int push(int x)
    {
    if (front = = -1)
        {
        p[++rear] = x;
         front++;
        }
      else
        p[++rear] = x;
      }
int pfront()
    {
    return p[front];
    }
int ppop()
    {
```

Sample input:

Enter the no. of variables - 5

1
2
3
4
5

Sample Output

Alternate elements in Queue

1
3
5

Explanation of Output: The output of this program is printed by taking the alternate elements not by divisible 2. By applying the conditions

```
{
 front++;
}
int pempty()
    {
    if (front > rear)
       return 1;
     else
      return 0;
   }
int main()
   {
    int n, i, x;
   print ("enter no of elements of Queue");
   Scanf("%d", &n);
   for (i=0; i<n; i++)
    {
     Scanf("%d", &x);
      p push(x);
    }
   i=0
   while (p empty() != 1)
      {
      if (i%2 == 0)
     printf("%d", p from());
      i++;
     p pop();
   }
```

5. i) problem:

How array is different from the linked list

* Array and linked list are used to store data of similar type
* In array we work with the index; where as on linked list we start from head(fix) and works by linking the next nodes.
*

PARAMETER	ARRAY	LINKEDLIST
Size	fixed	Variable
Memory allocation	Continous memory allocation	Random Memory allocation
process	Direct Access	Random Access Sequential
memory utilization	in efficient	efficient
Searching	Binary search, linear Seach etc	linear seach

* Insertion and deletion operation are easy in linked list whereas in arrays it is impossible
* Arrays depend on the index where as linked list relies on reference to previous and next element

iii. problem : Write a program to add the first element of one 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 list 1 and {5, 6} for list 2

Explanation of problem : The aim of the program is to create two linked lists and add the first element of second list to first list

Algorithm

*start
*Create two linked lists using two structs
*Write the conditions to enter the elements in list 1 & list 2
*write the condition to add 1st element in 2nd to 1st list
* print two lists 1 & 2
*stop.

Program in C

```
#include <stdio.h>
#include <stdlib.h>
  struct node {
        int data;
        struct node * link;
  }*head1 = NULL, * temp , * temp1 , *head2=NULL;
  Struct node *inseat (struct node *head, int x )
    {
```

```
temp1=(struct node *)malloc(sizeof(struct node));
temp1->data=x;
temp1->link=NULL;
if(head1==NULL)
 {
 head1=temp1;
 }
else
 {
temp1=head;
 while(temp1->link!=NULL)
  {
  temp1=temp1->link;
  }
  temp1->link=temp;
  }return head
}
int main()
 {
  int p,q,x,i;
  printf("enter no.of elements of first linked list");
  scanf("%d",&p);
  for(i=0;i<p;i++)
     {
     printf("enter the number");
     scanf("%d",&x);
     head1=insert(head1,x);
  }
```

```
printf("enter the no. of elements in second linkedlist");
scanf("%d", &q);
for(i=0; i<q; i++)
  {
   printf("Enter the element ");
   scanf("%d", &x);
   head2 = insert(head2, x);
  }
temp = (struct node *) malloc (sizeof(structnode));
 temp → link = head1;
 temp→data = head2 → data;
 head1 = temp;
 ~~temp~~ head2 = head2 → link;
  temp1 = head1;
while (temp1 != NULL)
   {
   printf("%d ", temp1 → data);
   temp1 = temp1 → link;
  }
printf ("\n linked list 2 \n");
 temp1 = head2;
while (temp1 != NULL)
    {
    printf("%d ", temp1 → data);
    temp1 = temp1 → link;
    }
```

Sample Input

Enter the no of elements of first linked list - 3

1
2
3

Enter the no of elements of second linked list - 3

4
5
6

Sample Output:

elements of linked list-1

4 1 2 3

elements of linked list-2

5. 6

Explanation of Output:

The output of this program is printed by creating two lists and by satisfying all the condition the 1st element of 2nd list is added to 1st list