DAY 3 ASSIGNMENT

1. Write a function “insert\_any()” for inserting a node at any given position of the linked list. Assume position starts at 0.

Ans)

#include<stdio.h>

#include<stdlib.h>

**void** insert\_any(**int**);

**void** create(**int**);

struct node

{

**int** data;

    struct node \*next;

};

struct node \*head;

**void** main ()

{

**int** choice,item,loc;

**do**

    {

        printf("\nEnter the item which you want to insert?\n");

        scanf("%d",&item);

**if**(head == NULL)

        {

            create(item);

        }

**else**

        {

            insert\_any(item);

        }

        printf("\nPress 0 to insert more ?\n");

        scanf("%d",&choice);

    }**while**(choice == 0);

}

**void** create(**int** item)

{

        struct node \*ptr = (struct node \*)malloc(sizeof(struct node \*));

**if**(ptr == NULL)

        {

            printf("\nOVERFLOW\n");

        }

**else**

        {

            ptr->data = item;

            ptr->next = head;

            head = ptr;

            printf("\nNode inserted\n");

        }

}

**void** insert\_any(**int** item)

    {

        struct node \*ptr = (struct node \*) malloc (sizeof(struct node));

        struct node \*temp;

**int** i,loc;

**if**(ptr == NULL)

        {

            printf("\nOVERFLOW");

        }

**else**

        {

            printf("Enter the location");

            scanf("%d",&loc);

            ptr->data = item;

            temp=head;

**for**(i=0;i<loc;i++)

            {

                temp = temp->next;

**if**(temp == NULL)

                {

                    printf("\ncan't insert\n");

**return**;

                }

            }

            ptr ->next = temp ->next;

            temp ->next = ptr;

            printf("\nNode inserted");

        }

}

Q) Write a function “delete\_beg()” for deleting a node from the beginning of the linked list.

Ans)

#include<stdio.h>

#include<stdlib.h>

**void** create(**int**);

**void** delete\_beg();

struct node

{

**int** data;

    struct node \*next;

};

struct node \*head;

**void** main ()

{

**int** choice,item;

**do**

    {

        printf("\n1.Append List\n2.Delete node\n3.Exit\n4.Enter your choice?");

        scanf("%d",&choice);

**switch**(choice)

        {

**case** 1:

            printf("\nEnter the item\n");

            scanf("%d",&item);

            create(item);

**break**;

**case** 2:

            delete\_beg();

**break**;

**case** 3:

            exit(0);

**break**;

**default**:

            printf("\nPlease enter valid choice\n");

        }

    }**while**(choice != 3);

}

**void** create(**int** item)

    {

        struct node \*ptr = (struct node \*)malloc(sizeof(struct node \*));

**if**(ptr == NULL)

        {

            printf("\nOVERFLOW\n");

        }

**else**

        {

            ptr->data = item;

            ptr->next = head;

            head = ptr;

            printf("\nNode inserted\n");

        }

    }

**void** delete\_beg()

    {

        struct node \*ptr;

**if**(head == NULL)

        {

            printf("\nList is empty");

        }

**else**

        {

            ptr = head;

            head = ptr->next;

            free(ptr);

            printf("\n Node deleted from the begining ...");

        }

}

Q) Write a function “delete\_end()” for deleting a node from the end of the linked list.

Ans)

#include<stdio.h>

#include<stdlib.h>

**void** create(**int**);

**void** end\_delete();

struct node

{

**int** data;

    struct node \*next;

};

struct node \*head;

**void** main ()

{

**int** choice,item;

**do**

    {

        printf("\n1.Append List\n2.Delete node\n3.Exit\n4.Enter your choice?");

        scanf("%d",&choice);

**switch**(choice)

        {

**case** 1:

            printf("\nEnter the item\n");

            scanf("%d",&item);

            create(item);

**break**;

**case** 2:

            end\_delete();

**break**;

**case** 3:

            exit(0);

**break**;

**default**:

            printf("\nPlease enter valid choice\n");

        }

    }**while**(choice != 3);

}

**void** create(**int** item)

    {

        struct node \*ptr = (struct node \*)malloc(sizeof(struct node \*));

**if**(ptr == NULL)

        {

            printf("\nOVERFLOW\n");

        }

**else**

        {

            ptr->data = item;

            ptr->next = head;

            head = ptr;

            printf("\nNode inserted\n");

        }

    }

**void** end\_delete()

    {

        struct node \*ptr,\*ptr1;

**if**(head == NULL)

        {

            printf("\nlist is empty");

        }

**else** **if**(head -> next == NULL)

        {

            head = NULL;

            free(head);

            printf("\nOnly node of the list deleted ...");

        }

**else**

        {

            ptr = head;

**while**(ptr->next != NULL)

                {

                    ptr1 = ptr;

                    ptr = ptr ->next;

                }

                ptr1->next = NULL;

                free(ptr);

                printf("\n Deleted Node from the last ...");

            }

        }