10.singly linked list :-

```
#include<stdio.h>
#include<malloc.h>
#include<stdlib.h>

struct node
{
	int info;
	struct node *link;
};

typedef struct node* NODE;

NODE getnode();
NODE insert_front(NODE , int);
NODE delete_front(NODE);
void display(NODE);

int main()
{
	NODE first;
	int choice, item;

	first = NULL;

	while(1)
	{
		printf("Enter\n");
		printf("1. Insert Front\n");
		printf("2. Delete Front\n");
		printf("3. Display the list\n");
```

```
                printf("4. Exit\n");

                scanf("%d", &choice);

                switch(choice)
                {
                        case 1:
                                printf("Enter item to be inserted\n");
                                scanf("%d", &item);
                                first = insert_front(first, item);

                                break;

                        case 2:
                                first = delete_front(first);

                                break;

                        case 3:
                                display(first);

                                break;

                        default:
                                exit(0);
                }
        }
}

NODE getnode()
```

```
{
        NODE x;

        x = (NODE) malloc(sizeof(struct node));

        if(x == NULL)
        {
                printf("Node creation error\n");
                return 0;
        }

        return x;
}


NODE insert_front(NODE first , int item)
{
        NODE temp;

        temp = getnode();
        temp->info = item;

        temp->link = first;

        return temp;
}

NODE delete_front(NODE first)
{
        NODE temp;
```

```
	if(first == NULL)
	{
		printf("Cannot delete. Empty List\n");
		return first;
	}

	temp = first;
	first = first->link;

	printf("Deleted node is %d\n", temp->info);
	free(temp);

	return first;
}

void display(NODE first)
{
	NODE temp;

	printf("Contents of linked list is:\n");

	if(first == NULL)
	{
		printf("Cannot print. Empty list\n");
		return;
	}

	temp = first;

	while(temp != NULL)
	{
```

```
            printf("%d\t", temp->info);
            temp = temp->link;
        }

        printf("\n");
}
```

Output :-