

**Insert at beginning**

```

void addNode_begin( int dataIn)
{
int found,success;

pPrev=NULL;

success=insertNode(dataIn);
if (success==1)   printf("Data Inserted Successfully\n");
else  printf("Out of Memory... \n");
}

```

**Insert at end**

```

void searchNode_end(int target)
{

pPrev=NULL;

pLoc=pList->head;

while(pLoc!=NULL)
{
    pPrev=pLoc;
    pLoc=pLoc->link;
}
}

```

**void addNode\_end( int dataIn)**

```

{
int found,success;

searchNode_end(dataIn);

success=insertNode(dataIn);
if (success==1)   printf("Data Inserted Successfully\n");
else  printf("Out of Memory... \n");
}

```



**Insert at a given position**

```
int searchNode_insert_pos(int target) /*NOTE target is position*/
{
int i;

if(target>pList->count+1) return 0;

pPrev=NULL;
pLoc=pList->head;

for(i=1;i<target;i++)
{
    pPrev=pLoc;
    pLoc=pLoc->link;
}
return 1;
}
```

```
-----
void addNode_pos( int dataIn, int pos)
{
int found,success;

found = searchNode_insert_pos (pos);
if (found==0) printf("Invalid position\n");
else
{
    success=insertNode(dataIn);
    if (success==1)    printf("Data Inserted Successfully\n");
    else    printf("Out of Memory... \n");
}
}
-----
```



**Insert before a given value**

```

int searchNode_insert_before_value(int target)
{
    /*NOTE target is list value before which insert*/
    int i;

    pPrev=NULL;
    pLoc=pList->head;

    while(pLoc!=NULL && target != pLoc->data)
    {
        pPrev=pLoc;
        pLoc=pLoc->link;
    }

    if (pLoc==NULL)
        return 0; /*Not found*/
    else
        if (target == pLoc->data) return 1; /*FOUND*/
    }
}
-----
void addNode_before_val( int dataIn, int dataRef)
{
    int found,success;

    found = searchNode_insert_before_value (dataRef);
    if (found==0) printf("Invalid Reference data\n");
    else
    {
        success=insertNode(dataIn);
        if (success==1)    printf("Data Inserted Successfully\n");
        else    printf("Out of Memory... \n");
    }
}
-----

```



**Insert after a given value**

```

int searchNode_insert_after_value(int target)
{
    /*NOTE target is list value after which insert*/
    int i;

    pPrev=NULL;
    pLoc=pList->head;

    while(pLoc!=NULL && target != pLoc->data)
    {
        pPrev=pLoc;
        pLoc=pLoc->link;
    }

    if (pLoc==NULL)
        return 0; /*Not found*/
    else
        if (target == pLoc->data)
        {
            pPrev=pLoc;
            pLoc=pLoc->link;
            return 1; /*FOUND*/
        }
}

```

```

-----
void addNode_after_value ( int dataIn, int dataRef)
{
    int found,success;

    found = searchNode_insert_after_value (dataRef);
    if (found==0) printf("Invalid Reference data\n");
    else
    {
        success=insertNode(dataIn);
        if (success==1)    printf("Data Inserted Successfully\n");
        else    printf("Out of Memory... \n");
    }
}

```

