

Linked Lists

Instructors:

Md Nazrul Islam Mondal &
Rizoan Toufiq

Department of Computer Science & Engineering
Rajshahi University of Engineering &
Technology Rajshahi-6204

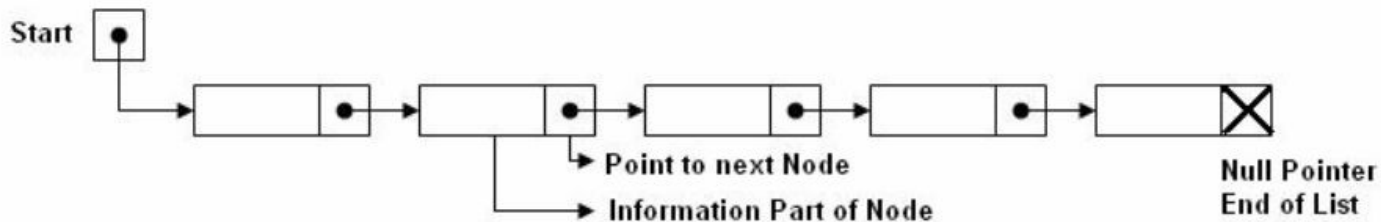
Outline

- **Introduction**
- **Linked List**
- **Representation of Linked Lists in Memory**
- Traversing a Linked List
- Searching a Linked List
- Memory Allocation; Garbage Collection
- Insertion into a Linked List
- Deletion from a Linked List
- Header Linked List
- Two Way Lists

Linked List

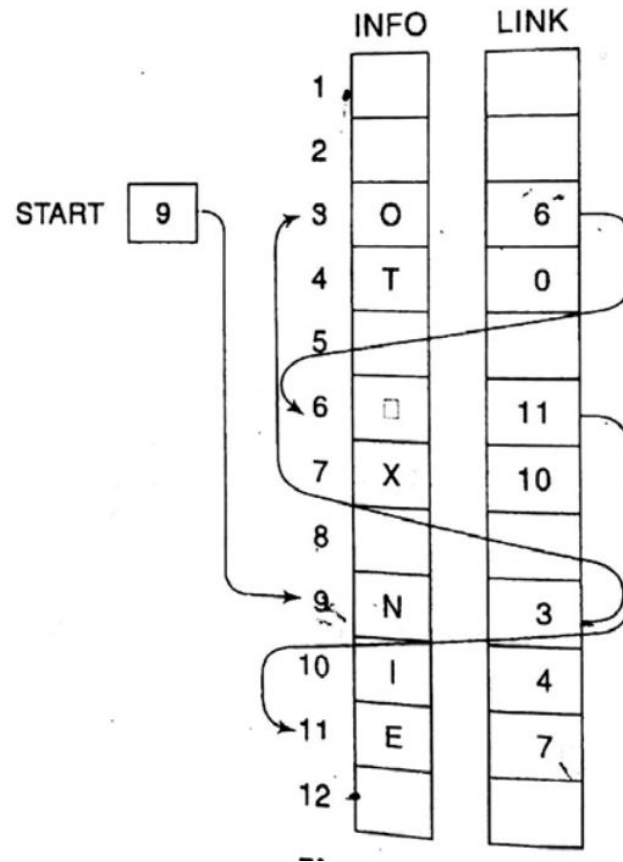
Linked List

- A linked list or one way list is a **linear** collection of data elements, called nodes, where the linear order is given by means of pointers.
- Each node is divided into **two** parts:
 - The first part - the **information** of the element/node
 - The second part- the address of the next node
- There is a special pointer **Start/List** contains the address of **first** node in the list.
- If this special **pointer** contains **null**, means that **List** is **empty**.

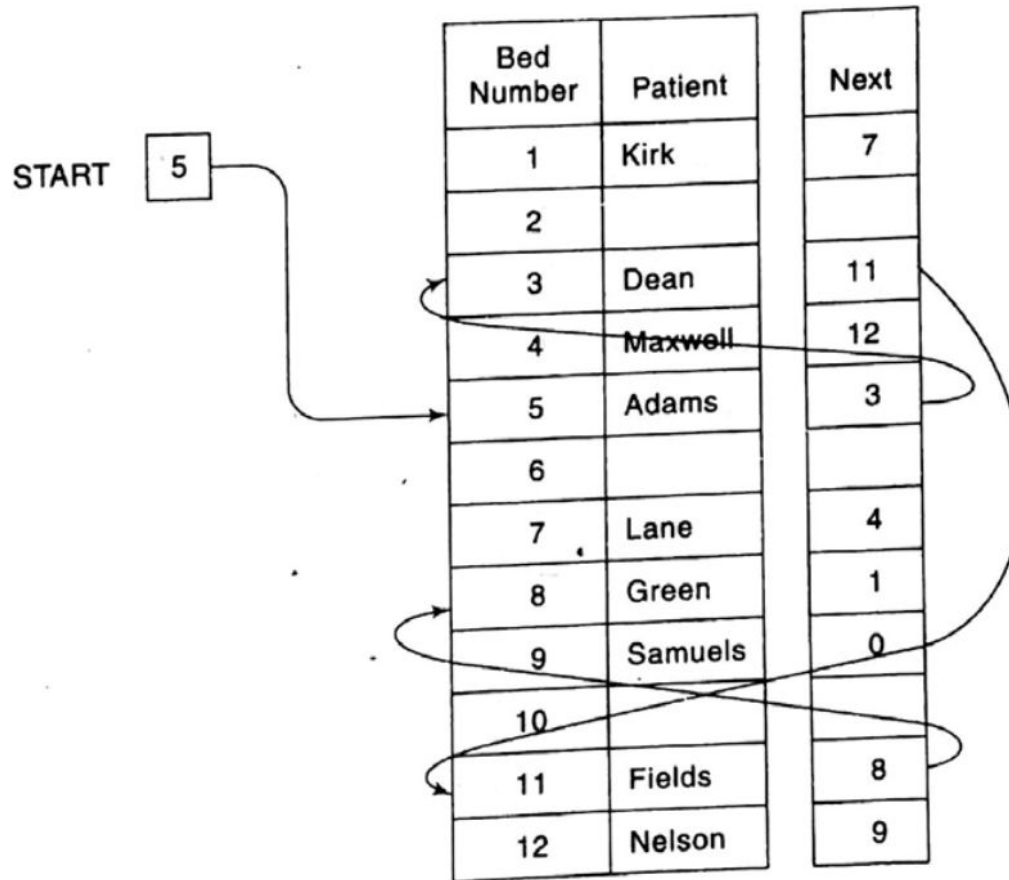


Representation of Linked Lists in Memory

Representation of Linked Lists in Memory



Representation of Linked Lists in Memory



Representation of Linked Lists in Memory

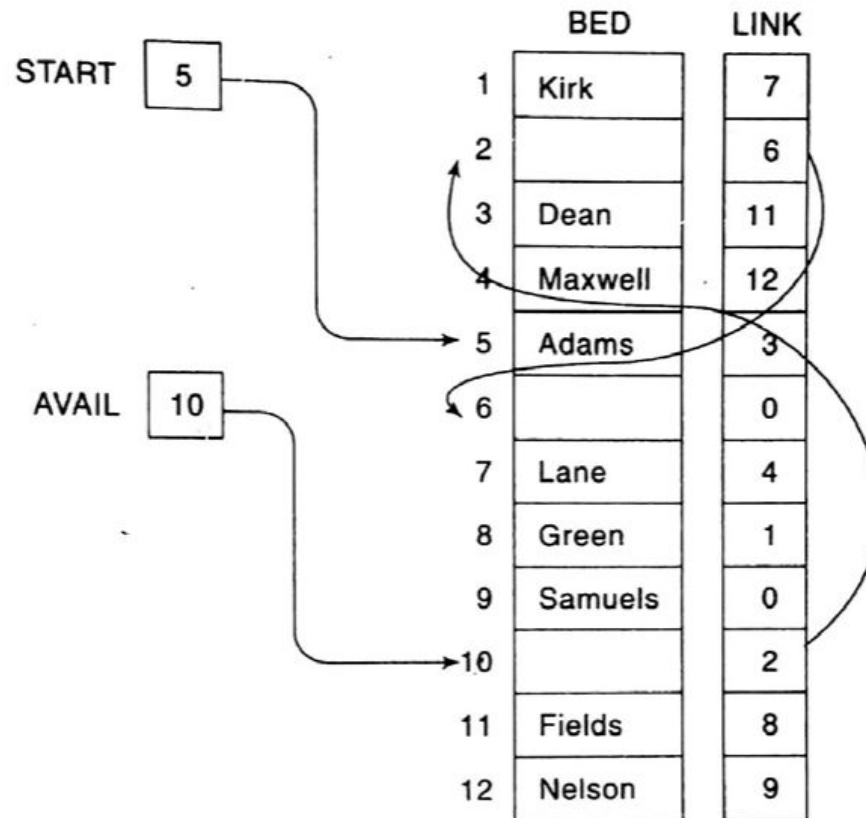
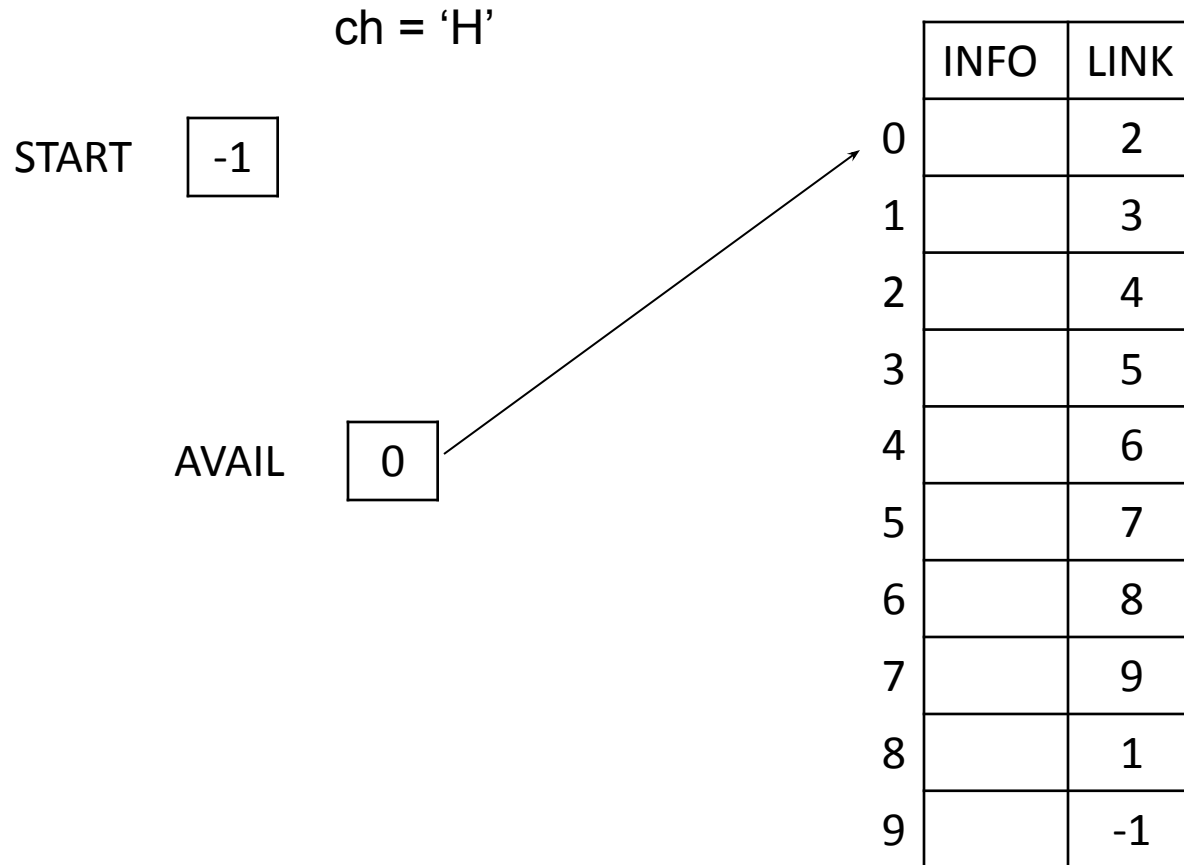
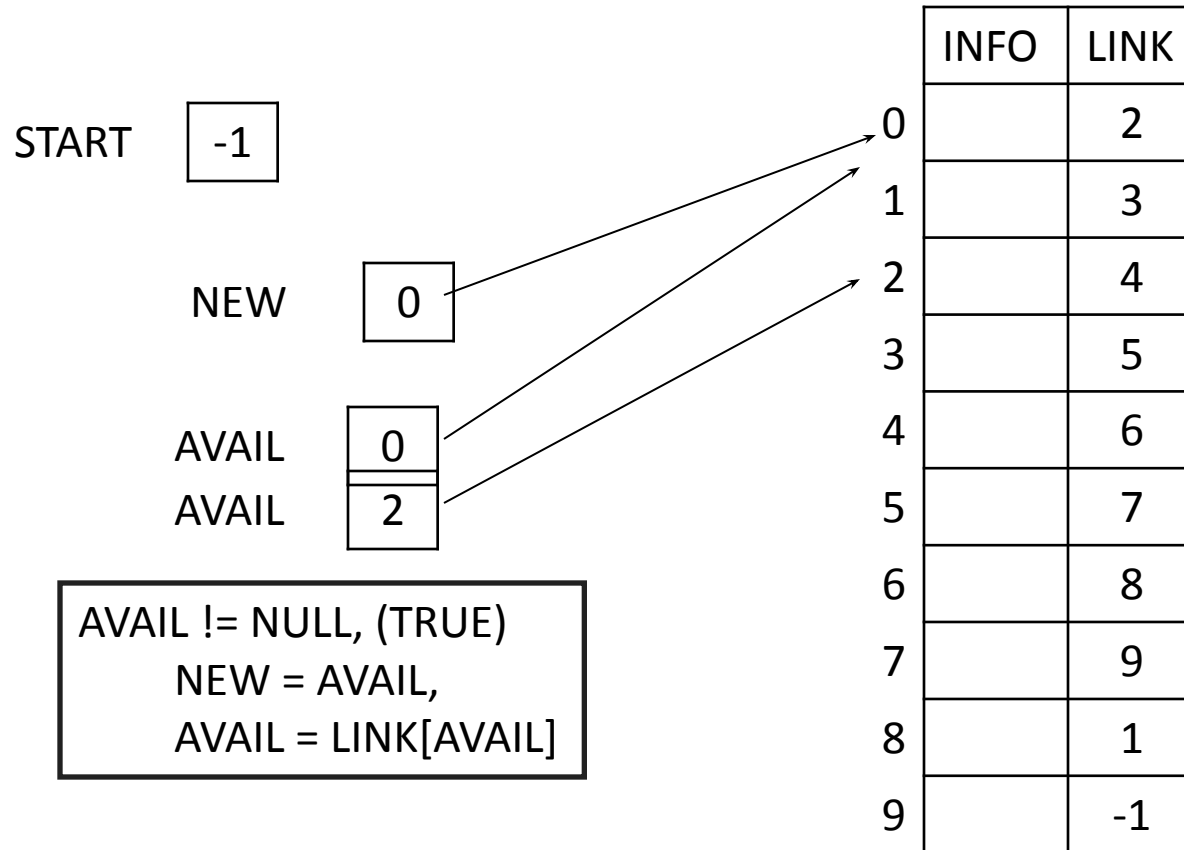


Fig. 5.9

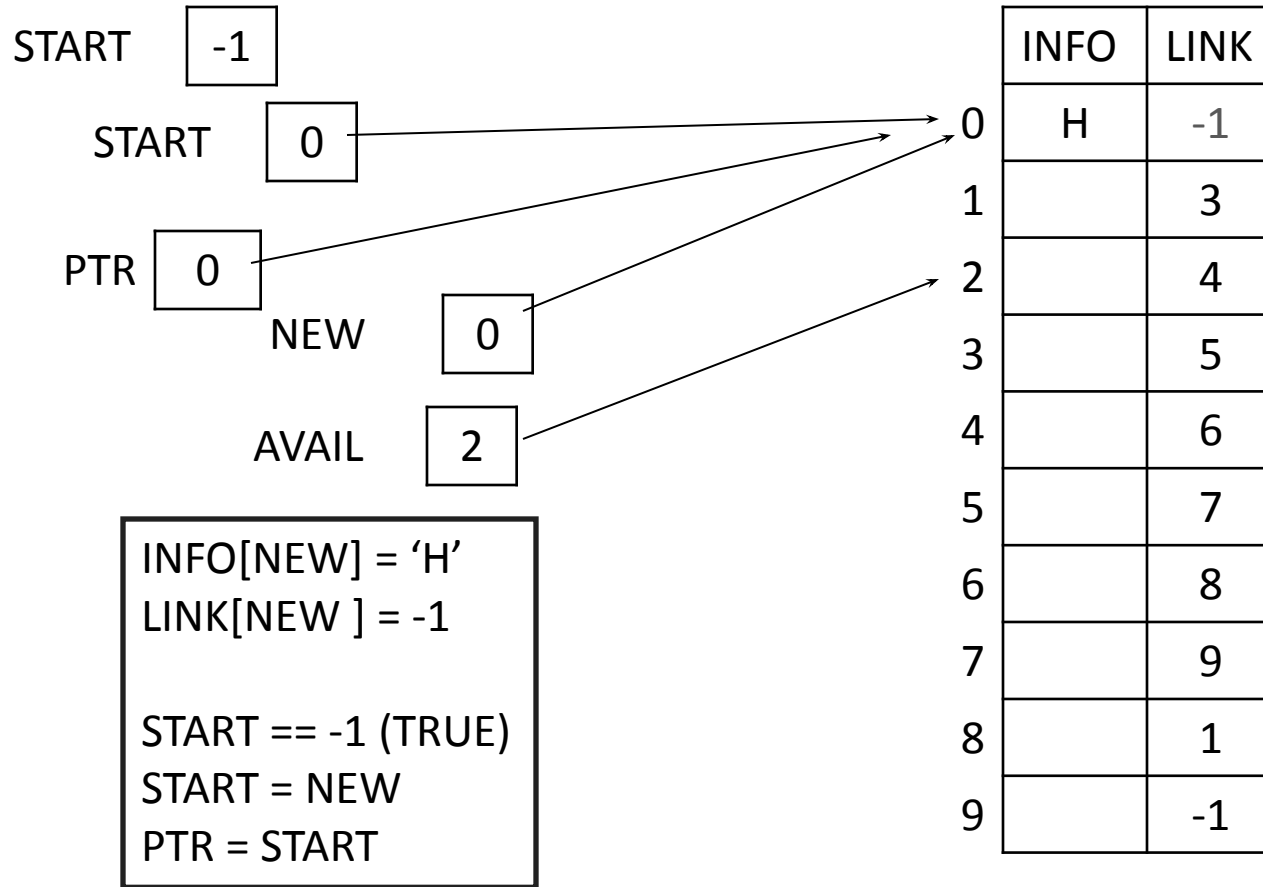
Create A Linked List



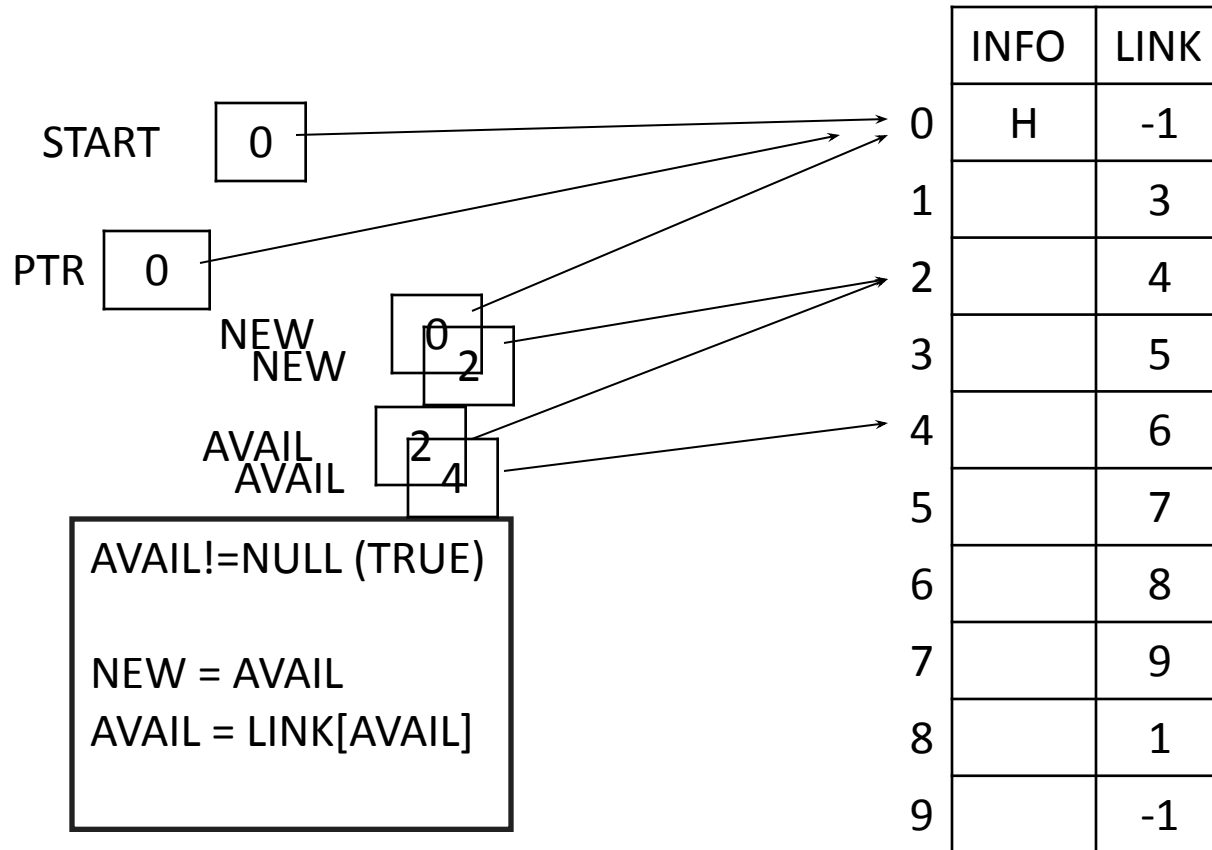
Create A Linked List



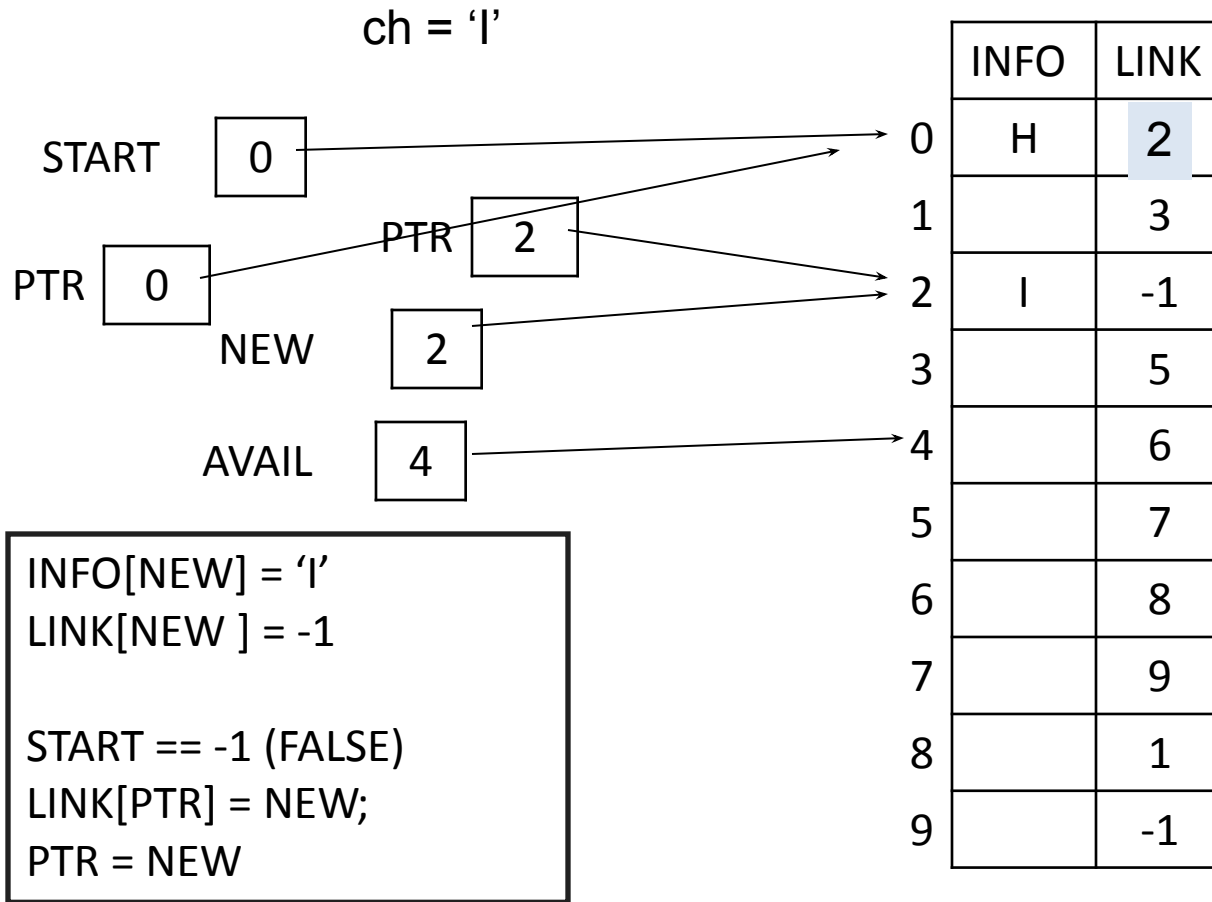
Create A Linked List



Create A Linked List



Create A Linked List



Available Memory Check

Algorithm: (Check Memory Overflow) This algorithm check available memory.

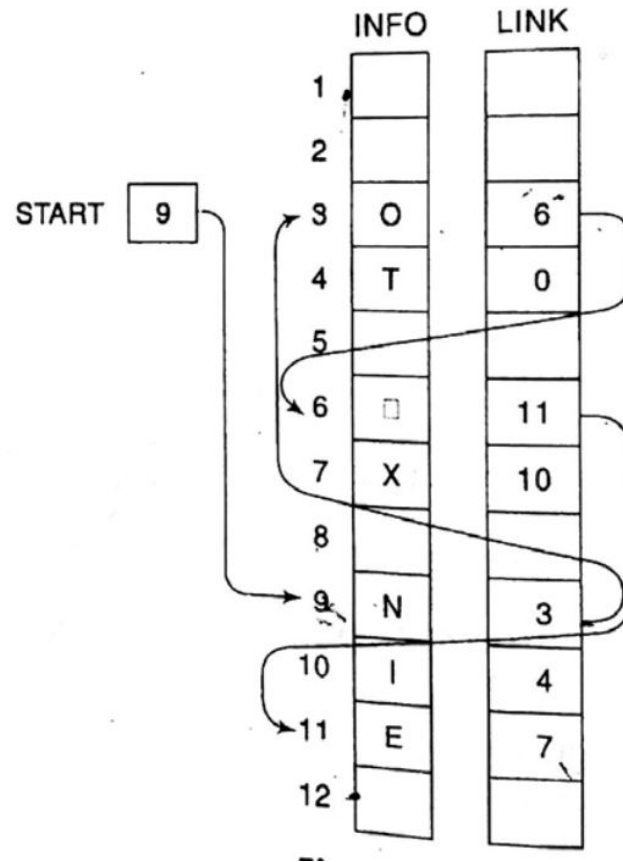
1. [OVERFLOW] If AVAIL = NULL, then: Write: OVERFLOW, and Exit
2. [Remove first node from AVAIL.]
Set NEW:=AVAIL and AVAIL := LINK[AVAIL]
3. Exit

Create A Linked List

Algorithm: (Create a Linked List) This algorithm create a linked list with n nodes.

1. START := NULL
2. Repeat Steps 3 to 5 for I = 1 to N
 [OVERFLOW] If AVAIL = NULL, then: Write:
 OVERFLOW, and Exit
3. [Remove first node from AVAIL.]
 Set NEW:=AVAIL and AVAIL := LINK[AVAIL]
4. Set INFO[NEW] := ITEM and LINK[NEW] := NULL
5. If START = NULL, then:
 Set START := NEW and PTR := START
 Else:
 Set LINK[PTR] := NEW and PTR = NEW
 [End of If structure]
6. Exit

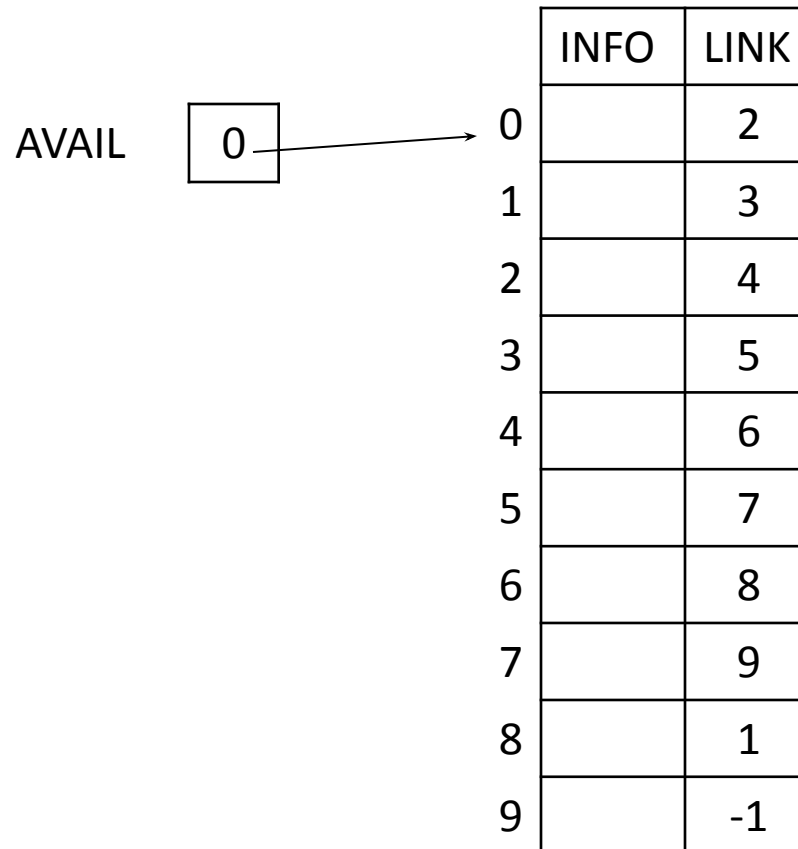
Create a Linked List Using Array



Create a Linked List Using Array

```
#include<stdio.h>
/*Parallel Global Array*/
char info[10]={};
int link[10]={2,3,4,5,6,7,8,9,1,-1};
int start = -1;
int avail = 0;
```

Create a Linked List Using Array



Create a Linked List Using Array

```
int newnode(){
    int newindex;
    if(avail == -1){
        printf("Overflow\n");
        return -1;
    }
    else{
        newindex = avail;
        avail = link[avail];
        return newindex;
    }
}
```

Create a Linked List Using Array

```
void creat_List(){  
    int ptr=-1,i,newindex; /*last node*/  
    char ch = 'H';
```

```
    for (i=0;i<8;i++){  
        newindex = newnode();  
        if (newindex ==-1)  
            break;
```

```
        info[newindex] = ch;  
        link[newindex] = -1;
```

```
        if(start == -1){  
            start = newindex;  
            ptr = newindex;  
        }  
        else{  
            link[ptr] = newindex;  
            ptr = newindex;  
        }  
        ch = ch + 1;  
    }  
    ptr = -1; /*Reset*/  
}
```

Create a Linked List Using Array

```
void traverse_List(){
    int ptr = 0;

    ptr = start;
    printf("Current\t Info \t Next\n");
    while (ptr!=-1){
        printf("%d \t %c \t %d \n",ptr,info[ptr], link[ptr]);
        ptr = link[ptr];
    }
}
```

Create a Linked List Using Array

```
int main(){  
    creat_List();  
    traverse_List();  
    return 0;  
}
```

Create a Linked List Using Structure

```
struct node{  
    char Info;  
    struct node *Link  
};
```



Create a Linked List Using structure

```
#include<stdio.h>
#include<stdlib.h>
struct node{
    char info;
    struct node *link;
};
struct node *start = NULL;
```


Create a Linked List Using Structure

```
void create_List(){
    struct node *ptr=NULL, *newnode = NULL;
    char ch = 'H';
    int i;
    for (i=0;i<4;i++){
        newnode = (struct node *)
        newnode->info = ch;
        newnode->link = NULL;
```

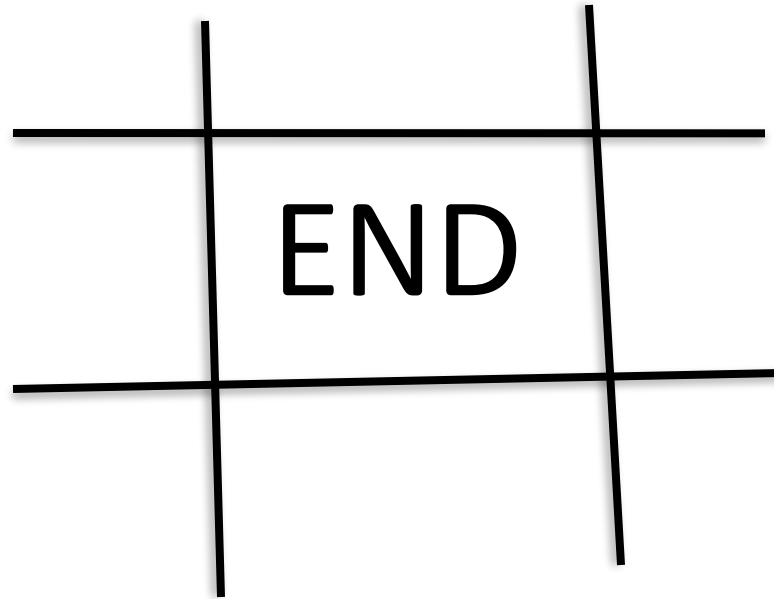
```
        if(start == NULL){
            start = newnode;
            ptr = newnode;
        }
        else{
            ptr->link = newnode;
            ptr = newnode;
        }
        ch = ch + 1;
    }
    ptr = NULL;
}
```

Create a Linked List Using Structure

```
void traverse_List(){
    struct node *ptr = start ;
    printf("Current\t Info \t Next\n");
    while (ptr!=NULL){
        printf("%x \t %c \t %x \n",ptr,ptr->info, ptr->link);
        ptr = ptr->link;
    }
}
```

Create a Linked List Using Structure

```
int main(){  
    creat_List();  
    traverse_List();  
    return 0;  
}
```



END