Neha B Chadaga
1BM19CS098

18/11/2020

WEEK 9: Program to implement singly linked list (insertion).

Algorithm:

Step 1: [Create struct node]

```
struct node
{
  int info;
  struct node *link;
}
typedef struct node *NODE;
```

Step 2: [Method to get a node]

```
NODE getnode()
{
  x = (NODE) malloc(sizeof(struct node));
  return x if x != NULL
}
```

Step 3: [Method to freenode]

```
NODE freenode()
  free(x);
```

Step 4: [Method to insert at front]

```
NODE insert_front()
{
  NODE temp;
  temp = getnode();
  temp->info = item;
```

```
temp -> link = NULL;
if(first == NULL)
    return temp;
temp -> link = first;
first = temp;
return first;
}
```

Step 5 : [Method to insert at rear]

```
NODE insert_rear()
{
    NODE temp, cur;
    temp = getnode();
    temp -> info = item
    temp -> link = NULL;
    if(first == NULL)
        return temp;
    cur = first;
    while(cur -> link != NULL)
        cur = cur -> link
    cur -> link = temp;
    return first;
}
```

Step 6 : [Method to insert at any position]

```
NODE insert_pos()
{
```

```
NODE temp, prev, cur;
 int count;
 temp=getnode();
 temp->info = item;
 temp-> link = NULL;
 if(first == NULL && pos == 1)
  return temp
 if(first == NULL)
 { print("Invalid position");
  return first; }
 if(pos == 1) {
  temp -> link = first
  first = temp;
  return temp;
   }
count = 1;
 prev=NULL;
  cur = first;
while(cur != NULL && count != pos)
  {
     prev = cur;
     cur = cur -> link;
     count ++;
 }
if(count == pos)
  {
```

```
prev->link = temp;
 temp -> link = cur;
return first;
}
printf("Invalid position");
return first;
}
```

Step 7: ~~CAtain method~~ [Display method]

```
~~NOL~~ void display()
        {
          NODE temp;
        if(first == NULL)
          print("List is empty");
        for(temp = first; temp != NULL; temp = temp->link)
          print("%d\n", temp->info);
      }
```

Step 8: [Main Method]

```
~~&~~ void main()
      {
        int item, choice, pos;
        NODE first = NULL;
        for(;;)
      {
        print("1. Insert front 2. Insert rear
              3. Insert at pos 4. Display 5. Exit");
        scanf(&choice);
```

RISH

```
switch(choice)
{
 case 1: print("Enter item to be inserted at front");
         scanf(item);
 case 2: print("Enter item to be inserted at rear");
         scanf(item);
 case 3: printf("Enter position and item to be
                 inserted");
         scanf(pos, item);
 Case 4: Display();
 case 5: Exit(0);
 }
}
```