① 17/01/2024 Linked list (insertion) week 3

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node * next;
};
struct Node * createNode(int value)
{
struct Node * new = (struct Node *) malloc (sizeof (struct Node));
if (newNode == NULL)
{
    printf("Memory allocation failed \n");
    exit(1)
}
newNode -> data = value;
newNode -> next = NULL;
return newNode;
}
struct Node * insertAtBeginning (struct Node * head, int value)
{
struct Node * newNode = createNode(value);
newNode -> next = head;
return newNode;
}
StructNode * insertAtanyPos (struct node * head, int value, int pos)
{
    struct Node * newNode = createNode(value);
    struct Node * temp = head;
    if (pos == 1)
    {
        newNode -> next = head;
    return newNode;
```

```
int main ()
struct node * head = NULL;
int choice, value, pos;
while (1)
{
    printf ("1 Insert at end \n");
    printf ("2 Insert at beginning \n");
    printf ("3 Insert at any position \n");
    printf ("4 Display list \n");
    printf ("Enter your choice: ");
    printf ("%d", &choice);
    switch (choice)
    {
    case 1: printf ("enter new value to be inserted")
            scanf ("%d", &value);
            head = insertAtend (head, value);
            break;
    case 2: printf ("enter the value to be inserted");
            scanf ("%d", &value);
            head = insertAtBeginning (head, value);
            break;
    case 3: printf ("Enter the value to be inserted")
            scanf ("%d", &value);
            printf ("Enter the position");
            scanf ("%d", &pos);
            head = insertAtanyPos (head, value, pos)
            break;
    case 4: displaylist (head);
            break;
```

(2)

```
int i;
for(i=1, i<pos-1, i++)
    temp = temp -> next;
}
newNode -> next = temp -> next;
temp -> next = newNode;
return head;
}
struct Node * insertAtend (structNode * head, int value)
{
    structNode * newNode = creatNode (value);
    if (head == NULL)
    {
        return newNode;
    }
    struct Node * temp = head;
    while (temp -> next != NULL)
    {
        temp = temp -> next
    }
    temp -> next = newNode;
    return head;
}
void displayList (structNode * head)
{
    structNode * temp = head;
    while (temp != NULL)
    {
        printf ("%d", temp -> data);
        temp = temp -> next;
    }
    printf ("NULL\n");
}
```

(3)

```
case 5 : exit(0)
default : printf("Invalid choice");
}
}
return 0;
}
```

→ output

1) Insert at end
2) Insert at beginning
3) Insert at any position
4) Display
5) Exit

Enter your choice 2
Enter the value to be inserted : 12

Enter your choice : 2
Enter the value to be inserted : 13

Enter your choice : 4
13 → 12 → NULL

Enter your choice : 1
Enter the value to be inserted : 23
Enter your choice : 4
13 → 12 → 23 → NULL

Enter your choice : 3
Enter the value to be inserted : 12
Enter the position to insert : 3
Enter your choice : 4
13 → 12 → 12 → 23 → NULL