• Reverse, operations on Singly Linked List and Sort, Concatenation

29.01.2024

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

Struct Node
{ int data;
  Struct Node* next; };

Void insert (Struct Node** head, int data)
{ struct Node *newnode = (struct Node*) malloc
                         (sizeof (struct Node));

  newnode -> data = data;
  newnode -> next = *head;
  *head = newnode;
}
Void print (struct Node *head)
{  while (head != NULL)
   { printf ("%d", head -> data);
     head = head -> next;
   } printf ("\n");
}
Void sort (Struct Node** head)
{ struct Node * current, *nextnode;
   int temp;
   while ( current != NULL)
      nextnode = current -> next;
      while (nextnode != NULL)
       { if (current -> data > nextnode -> data)
           {  temp = current -> data;
              current -> data = nextnode -> data;
              nextnode -> data = temp; }
          nextnode = nextnode -> next ; }
      current = current -> next; }
}
```

```
void reverselist (struct Node ** head)
{ struct Node *prev, *current, *nextnode;
  prev = NULL;
  current = *head;
  while (current != NULL)
  {  nextnode = current -> next;
     current -> next = prev;
     prev = current;
     current = next Node;
  } *head = prev;

void concatenatelists (struct Node ** list1, struct Node *list2)
{  if (*list1 == NULL)
     {  *list1 = list2;
        return; }
   struct Node *temp = *list1;
   while (temp -> next != NULL)
      temp = temp -> next;
   temp -> next = list2; }

void main () {
   struct Node * list1 = NULL;
   struct Node * list2 = NULL;
   int data, choice;
   while (1)
{  printf ("\n 1. Insert into List 1 \n");
   printf ("\n 2. Insert into List 2 \n");
   printf ("3. Sort List \n");
   printf ("4. Reverse List \n");
   printf ("5. Concatenate Lists \n");
   printf ("6. Print Lists \n");
   printf ("7. Exit \n");
   printf ("Enter your choice");
   scanf ("%d", &choice);
```

```
switch (choice)
{ case 1: printf ("Enter the data:");
          scanf ("%d", &data);
          insert (&list1, data);
          break;
  case 2: printf ("Enter the data");
          scanf ("%d", &data);
          insert (&list2, data);
          break;
  case 3: sort (&list1);
          printf ("List is sorted \n");
          break;
  case 4: reverse_list (&list1);
          printf ("List is reversed \n");
          break;
  case 5: concatenate lists (&list1, &list2);
          printf ("Lists are concatenated \n");
          break;
  case 6: printf ("List1: ");
          print (list1);
          printf ("List2:");
          print (list2);
          break;
  case 7: exit(0);
  default: printf ("Invalid input \n"); }
}}
```

→
1. Insert into List 1
2. Insert into list 2.
3. Sort the list
4. Reverse the list
5. Concatenate lists
6. Print lists
7. Exit.

Enter your choice: 1
Enter data to insert into list 1: 10

Enter your choice : 1.
Enter data to insert into list 1: 8
Enter your choice : 6
List 1 : 8 10
List 2 :
Enter your choice : 3
List1 Sorted.
Enter your choice : 4
List1 rerersed.
Enter your choice : 6.
List1 : 10 8
List 2 :
Enter your choice : 2.
Enter data to insert into list 2 : 20.
Enter your choice : 2
Enter data to insert into list 2 : 30.
Enter your choice : 6
List 1 : 10 8
List 2 : 30 20
Enter your choice : 5
Lists concatenated.
Enter your choice : 6.
List 1 : 10 8 30 20
List 2 : 30 20
Enter your choice : 7.

```
1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 1
Enter data to insert into List 1: 10

1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 1
Enter data to insert into List 1: 12

1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 6
List 1: 12 10
List 2:

1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 3
```

```
Enter your choice: 3
List 1 sorted.

1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 6
List 1: 10 12
List 2:

1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 4
List 1 reversed.

1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 6
List 1: 12 10
List 2:

1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
```

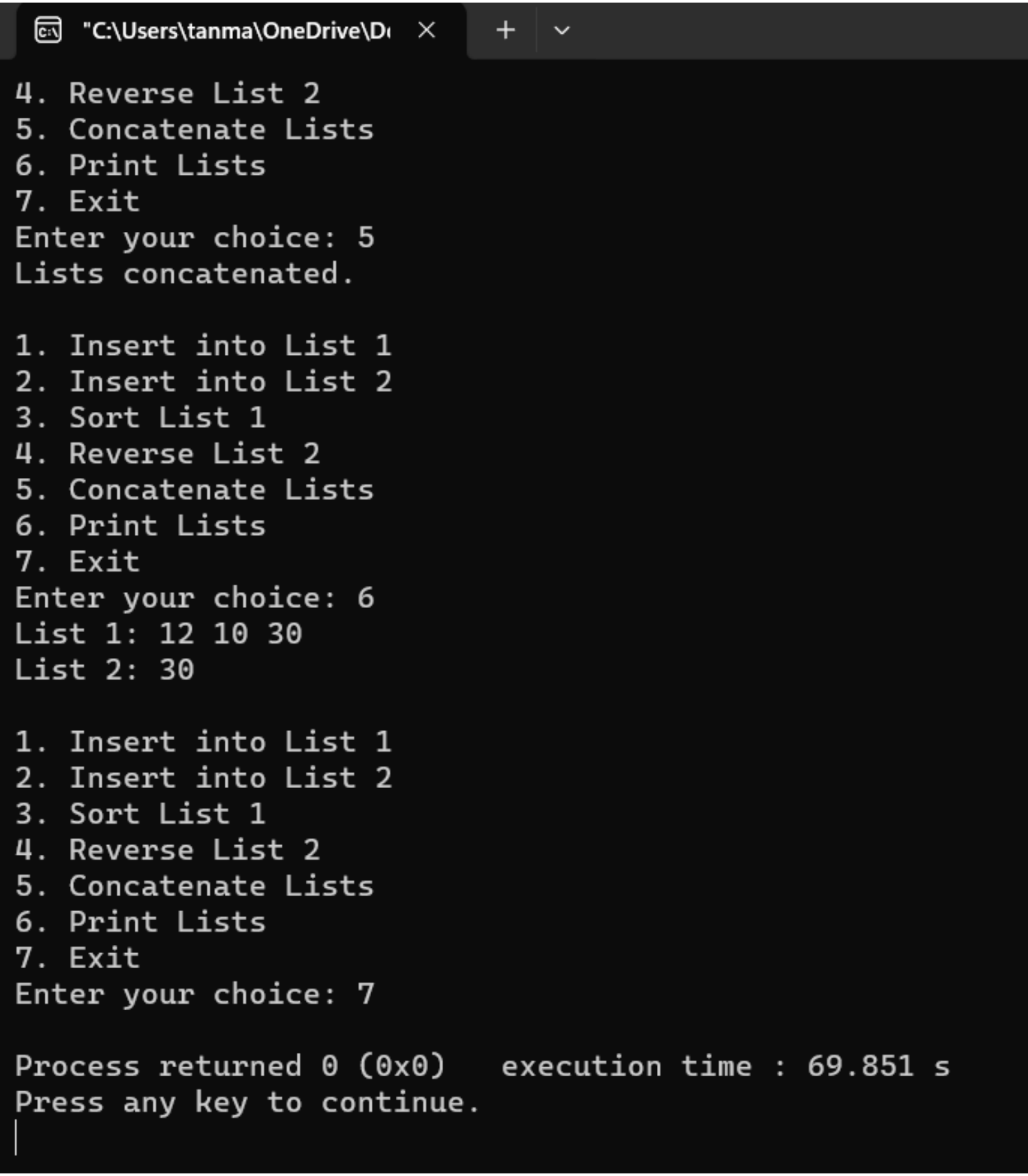
```
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 2
Enter data to insert into List 2: 30

1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 6
List 1: 12 10
List 2: 30

1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 5
Lists concatenated.

1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 6
List 1: 12 10 30
List 2: 30

1. Insert into List 1
2. Insert into List 2
```


"C:\Users\tanma\OneDrive\D
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 5
Lists concatenated.
1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 6
List 1: 12 10 30
List 2: 30
1. Insert into List 1
2. Insert into List 2
3. Sort List 1
4. Reverse List 2
5. Concatenate Lists
6. Print Lists
7. Exit
Enter your choice: 7
Process returned 0 (0x0) execution time : 69.851 s
Press any key to continue.