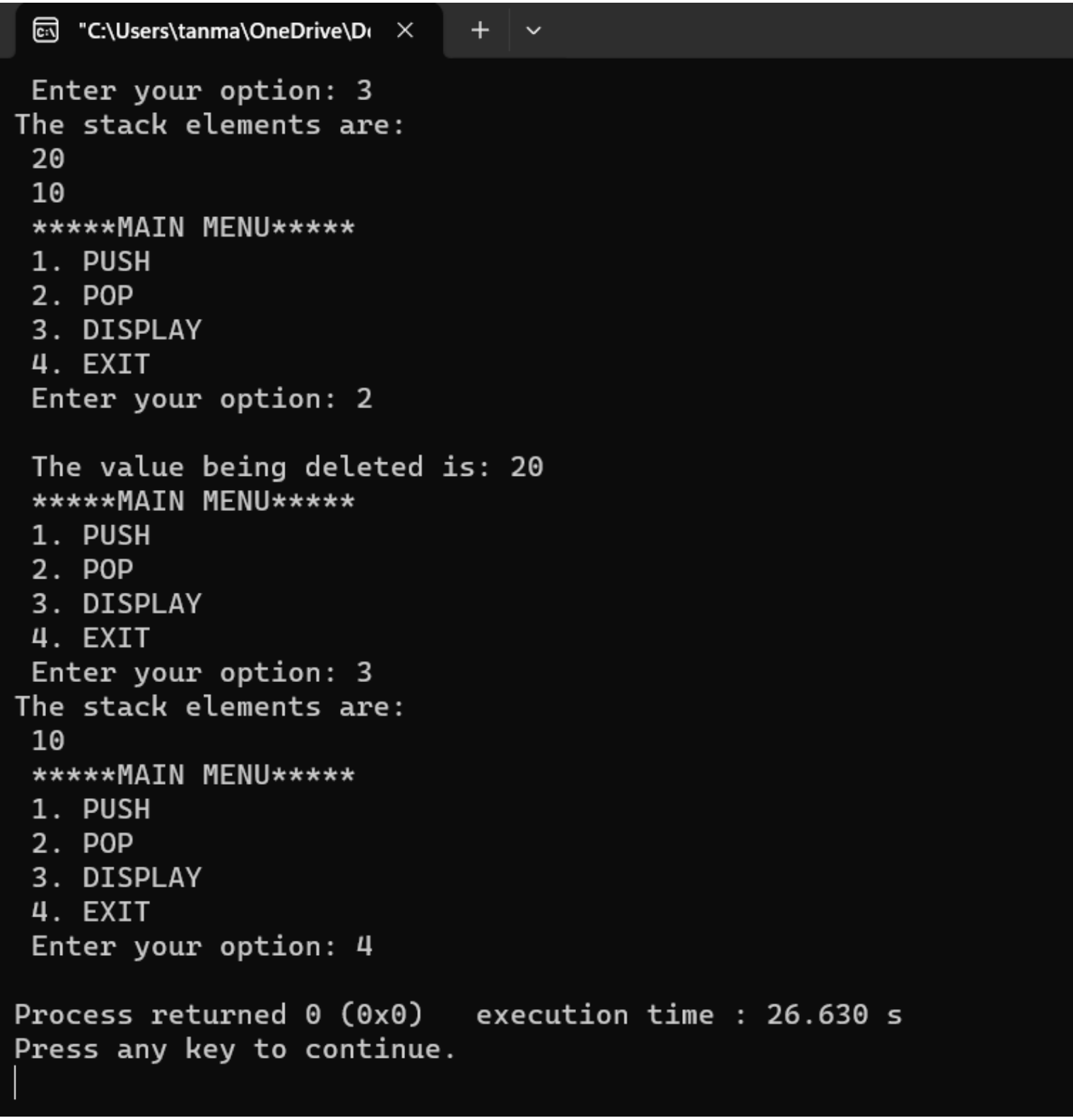
Stack implementation using linked lists 29.01.2024

```
#include <stdio.h>
#include <stdlib.h>
Struct stack
{ int data;
  struct stack * next;
};
struct stack *top = NULL;
struct stack * push (struct stack *, int val);
struct stack * pop ( struct stack*) ;
struct stack * display (struct stack *);
void main()
{ int val, option;
  while(1)
{ printf (" \n ** MENU** ");
  printf (" \n 1. PUSH");
  printf (" \n 2. POP");
  printf (" \n 3. DISPLAY");
  printf (" \n 4. EXIT");
  printf (" Enter your choice");
  Scanf ("%d", &option);
  Switch (option)
  { case 1: printf (" \n Enter the element to be inserted:");
            Scanf (" %d", &val);
            top = push (top, val);
            break;
    case 2: top = pop (top);
            break;
    case 3: top = display (top);
            break;
    case 4: exit(0);
    default: printf (" Invalid input...");
}}}
```

Ent
N
29/1/24

```
Struct Stack *push (struct stack * top, int val).
{ Struct stack *ptr;
  ptr = (struct stack *) malloc (sizeof (struct stack));
  ptr -> data = val;
  if (top == NULL)
  { ptr -> next = NULL;
    top = ptr;
    printf (" The value %d is inserted", val);
  }
  else {
       ptr -> next = top;
       top = ptr;
       printf (" The value %d is inserted", val);
    }
  return top; }

Struct stack * pop (struct stack* top).
{ Struct stack *ptr;
  ptr = top;
  if (top == NULL)
    printf ("\n Stack is Empty");
  else { while (ptr != NULL)
       { top = top -> next;
         printf (" The value deleted is : %d", ptr -> data);
         free (ptr); }
       return top; }

Struct Stack * display (struct stack * top)
{ Struct Stack * ptr;
  ptr = top;
  if (top == NULL)
    printf (" Stack is Empty");
  else { while (ptr != NULL)
       { printf (" The Stack elements are");
         printf ("\n %d", ptr -> data);
         ptr = ptr -> next; } }
    return top;
}
```

```
"C:\Users\tanma\OneDrive\D

 *****MAIN MENU*****
 1. PUSH
 2. POP
 3. DISPLAY
 4. EXIT
 Enter your option: 1

 Enter the number to be pushed on stack: 10
The value 10 is inserted
 *****MAIN MENU*****
 1. PUSH
 2. POP
 3. DISPLAY
 4. EXIT
 Enter your option: 1

 Enter the number to be pushed on stack: 20
The value 20 is inserted
 *****MAIN MENU*****
 1. PUSH
 2. POP
 3. DISPLAY
 4. EXIT
 Enter your option: 3
The stack elements are:
 20
 10
 *****MAIN MENU*****
 1. PUSH
 2. POP
 3. DISPLAY
 4. EXIT
 Enter your option: 2

 The value being deleted is: 20
 *****MAIN MENU*****
 1. PUSH
 2. POP
 3. DISPLAY
```


"C:\Users\tanma\OneDrive\D
Enter your option: 3
The stack elements are:
20
10
*****MAIN MENU*****
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Enter your option: 2
The value being deleted is: 20
*****MAIN MENU*****
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Enter your option: 3
The stack elements are:
10
*****MAIN MENU*****
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Enter your option: 4
Process returned 0 (0x0) execution time : 26.630 s
Press any key to continue.

Output:

** MENU **
1. PUSH 2. POP 3. DISPLAY 4. EXIT.

- Enter your option: 1
  Enter number: 10
  The value 10 inserted
- Enter your option: 3
  The stack elements are: 20 10
- Enter your option: 3
  The stack elements: 10
- Enter your option: 4

- Enter your option: 1
  Enter number: 20
  The value 20 inserted.
- Enter your option: 2
  The ~~stack~~ value deleted is 20

Queue implementation using Linked Lists

29-01-2024

```
#include<stdio.h>
#include<stdlib.h>
struct node { int data;
              struct node *next; };
struct queue { struct node * front;
               struct node * rear; };
struct queue *createqueue()
{ struct queue *q = (struct queue *) malloc(sizeof(struct queue));
   q -> front = q -> rear = NULL;
   return q;
}
struct queue *q;
struct queue * insert (struct queue*, int val);
struct queue * delete_ele ( struct queue *);
struct queue * display (struct queue *);
void main()
{ int val, choice;
 q=createqueue(q);
  while(1){
     printf("** MENU **");
     printf("\n 1. INSERT 2. DELETE. \n 3. DISPLAY, \n4.EX
     printf("\n Enter your option");
     scanf("%d", &choice);
```

```
Switch (option)
{ case 1: printf("\n Enter the number to be inserted in the queue");
          scanf("%d", &val);
          q = insert(q, val);
          break;
  case 2: q = delete_element(q)
          break;
  case 3: q = display(q)
          break;

  case 4: exit(0);
  default: printf(" Invalid input....."); }
}}

Struct queue *insert (struct queue *q, int val)
{  struct node *ptr;
   ptr = (struct node *) malloc (sizeof (struct node));
   ptr -> data = val;
   if (q -> front == NULL)
   {  q -> front = q -> rear = ptr;
      q -> front -> next = q -> rear -> next = NULL; }
   else {  q -> rear -> next = ptr;
           q -> rear = ptr;
           q -> rear -> next = NULL; }
     return q; }

Struct queue * display (struct queue *q)
{  struct node *ptr;
   ptr = q -> front;
   if (ptr == NULL)
     printf("\n Queue is empty*\n");
   else { while (ptr != q -> rear)
          { printf("%d\t", ptr -> data);
            ptr = ptr -> next; }
        printf("%d\t", ptr -> data); }
   return q;
}
```

```
Struct queue *delete_element (struct queue *q)
{ struct node *ptr;
  ptr = q -> front;
  if (q -> front == NULL)
    printf ("Queue is Empty");
  else { q -> front = q -> front -> next;
         printf (" The value deleted is : %d", ptr -> data;
         free(ptr); }
  return q;
}
```

Output :

```
** MENU **
1. Insert
2. Delete
3. Display
4. Exit.
```

- Enter your option : 1 , Enter the number : 10

  The value 10 is inserted.

- Enter your option : 1 , Enter the number : 20

  The value 20 is inserted

- Enter your option : 1 , Enter the number : 30

  The value 30 is inserted.

- Enter your option : 3

  10 20 30

- Enter your option : 2

  The value deleted is : 10

- Enter your option : 2

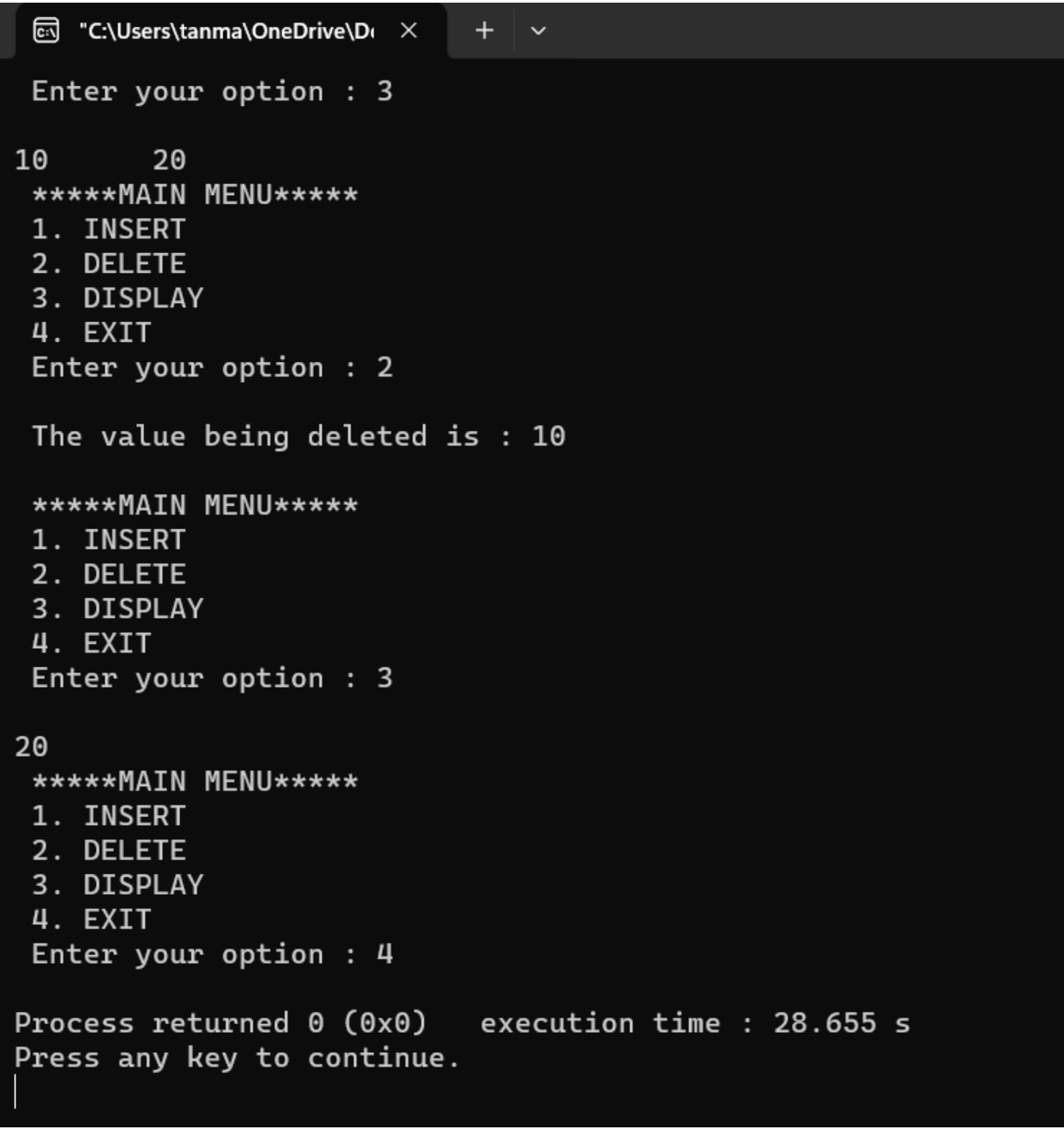
  The value deleted is : 20

- Enter your option : 3

  30

- Enter your option : 4

```
"C:\Users\tanma\OneDrive\D

 *****MAIN MENU*****
 1. INSERT
 2. DELETE
 3. DISPLAY
 4. EXIT
 Enter your option : 1

 Enter the number to insert in the queue:10

The value 10 is inserted into the queue.

 *****MAIN MENU*****
 1. INSERT
 2. DELETE
 3. DISPLAY
 4. EXIT
 Enter your option : 1

 Enter the number to insert in the queue:20

The value 20 is inserted into the queue.

 *****MAIN MENU*****
 1. INSERT
 2. DELETE
 3. DISPLAY
 4. EXIT
 Enter your option : 3

10      20
 *****MAIN MENU*****
 1. INSERT
 2. DELETE
 3. DISPLAY
 4. EXIT
 Enter your option : 2

 The value being deleted is : 10
```


"C:\Users\tanma\OneDrive\D
Enter your option : 3
10 20
*****MAIN MENU*****
1. INSERT
2. DELETE
3. DISPLAY
4. EXIT
Enter your option : 2
The value being deleted is : 10
*****MAIN MENU*****
1. INSERT
2. DELETE
3. DISPLAY
4. EXIT
Enter your option : 3
20
*****MAIN MENU*****
1. INSERT
2. DELETE
3. DISPLAY
4. EXIT
Enter your option : 4
Process returned 0 (0x0) execution time : 28.655 s
Press any key to continue.