29/1/2 **STACK USING LINKED LIST** NAME:Shivaraj K Pujari ,USN:1BM22CS259

```
#include <stdio.h>
#include <stdlib.h>
void push();
void pop();
void display();
struct node
{
int val;
struct node *next;
};
struct node *head;

void main ()
{
   int choice=0;
   printf("MENU: 1.push 2.pop 3.display 4.exit\n");
   while(choice != 4)
   {
      printf("Enter your choice \n");
      scanf("%d",&choice);
      switch(choice)
      {
         case 1:
         {
            push();
            break;
         }
         case 2:
         {
            pop();
            break;
         }
```

```
        case 3:
        {
            display();
            break;
        }
        case 4:
        {
            printf("Exiting");
            break;
        }
        default:
        {
            printf("Please Enter valid choice ");
        }
    };
}
}
void push ()
{
    int val;
    struct node *ptr = (struct node*)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("not able to push the element\n");
    }
    else
    {
        printf("Enter the value\n");
        scanf("%d",&val);
        if(head==NULL)
        {
            ptr->val = val;
            ptr -> next = NULL;
```

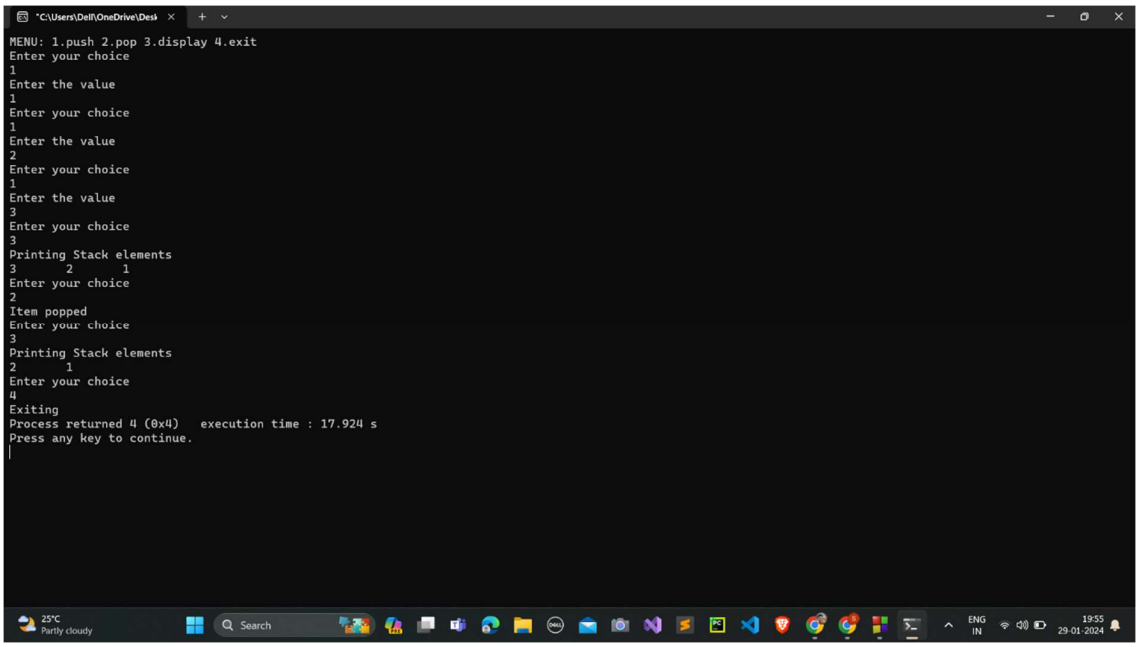
```
            head=ptr;
        }
        else
        {
            ptr->val = val;
            ptr->next = head;
            head=ptr;
        }
    }
}

void pop()
{
    int item;
    struct node *ptr;
    if (head == NULL)
    {
        printf("Underflow\n");
    }
    else
    {
        item = head->val;
        ptr = head;
        head = head->next;
        free(ptr);
        printf("Item popped\n");

    }
}
void display()
{
    int i;
    struct node *ptr;
```

```
    ptr=head;
    if(ptr == NULL)
    {
        printf("Stack is empty\n");
    }
    else
    {
        printf("Printing Stack elements \n");
        while(ptr!=NULL)
        {
            printf("%d\t",ptr->val);
            ptr = ptr->next;
        }
        printf("\n");
    }
}
```

OUTPUT:



QUEUES USING LINKEDLIST

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   int data;
   struct node *next;
};
struct node *front;
struct node *rear;
void insert();
void delete();
void display();
void main ()
{
   int choice;
   printf("MENU: 1.enqueue 2.dequeue 3.display 4.exit\n");
   while(choice != 4)
   {
      printf("Enter your choice \n");
      scanf("%d",& choice);
      switch(choice)
      {
         case 1:
         insert();
         break;
         case 2:
         delete();
         break;
         case 3:
         display();
         break;
         case 4:
```

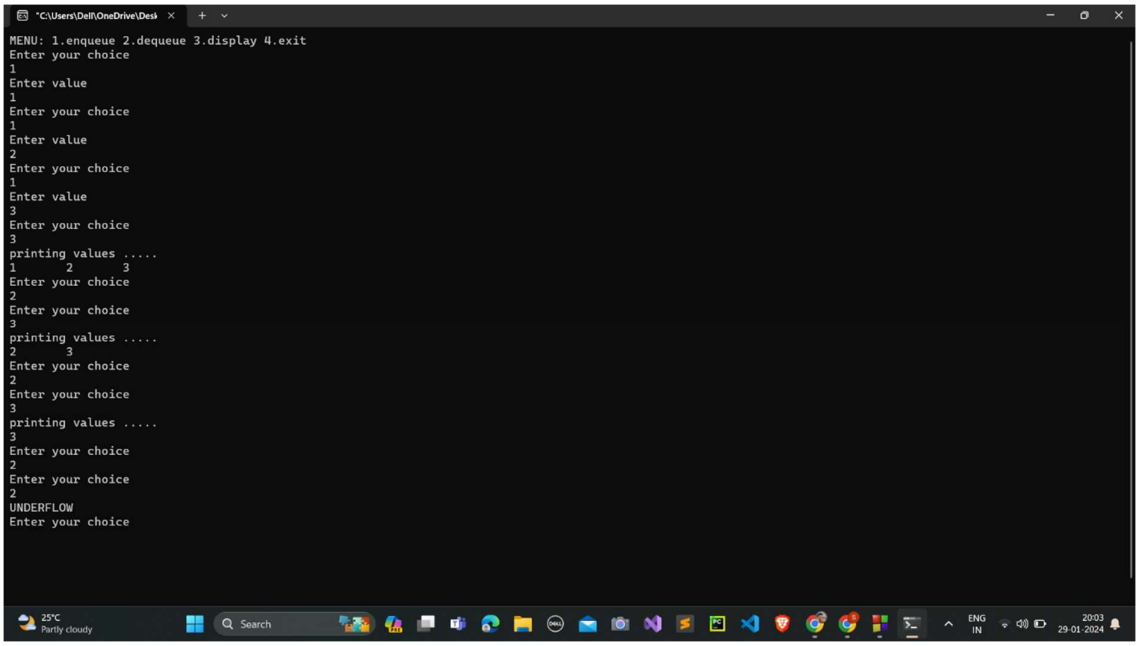
```
            exit(0);
            break;
            default:
            printf("Enter valid choice\n");
        }
    }
}
void insert()
{
    struct node *ptr;
    int item;

    ptr = (struct node *) malloc (sizeof(struct node));
    if(ptr == NULL)
    {
        printf("OVERFLOW\n");
        return;
    }
    else
    {
        printf("Enter value\n");
        scanf("%d",&item);
        ptr -> data = item;
        if(front == NULL)
        {
            front = ptr;
            rear = ptr;
            front -> next = NULL;
            rear -> next = NULL;
        }
        else
        {
            rear -> next = ptr;
```

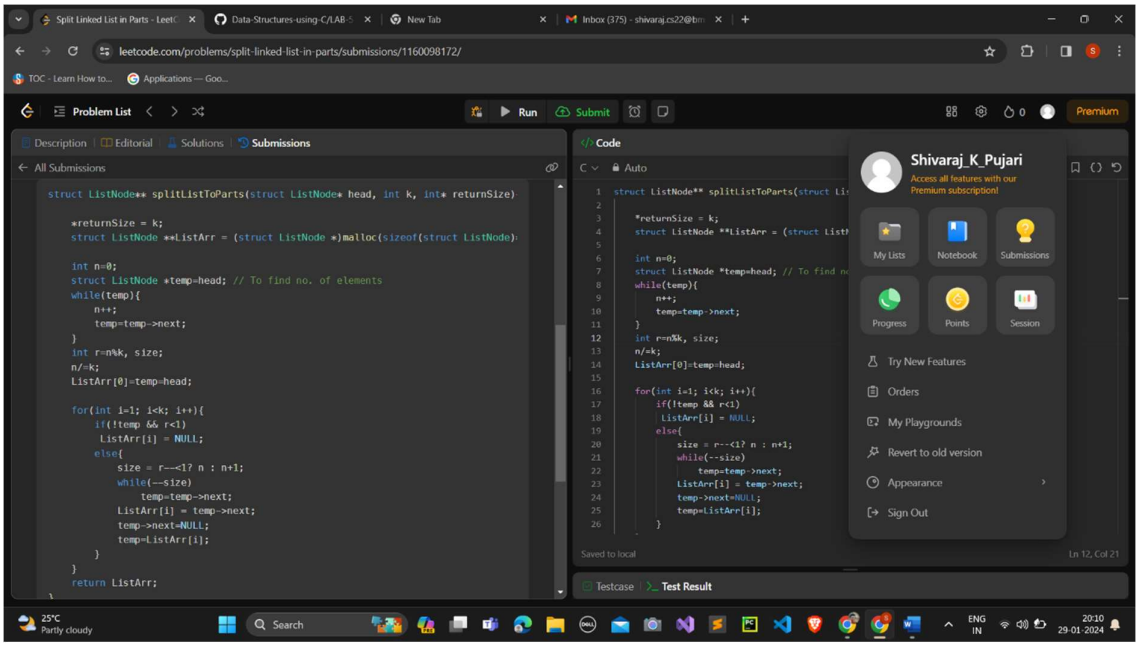
```
            rear = ptr;
            rear->next = NULL;
        }
    }
}
void delete ()
{
    struct node *ptr;
    if(front == NULL)
    {
        printf("UNDERFLOW\n");
        return;
    }
    else
    {
        ptr = front;
        front = front -> next;
        free(ptr);
    }
}
void display()
{
    struct node *ptr;
    ptr = front;
    if(ptr== NULL)
    {
        printf("Empty queue\n");
    }
    else
    {   printf("printing values .....\n");
        while(ptr != NULL)
        {
            printf("%d\t",ptr -> data);
```

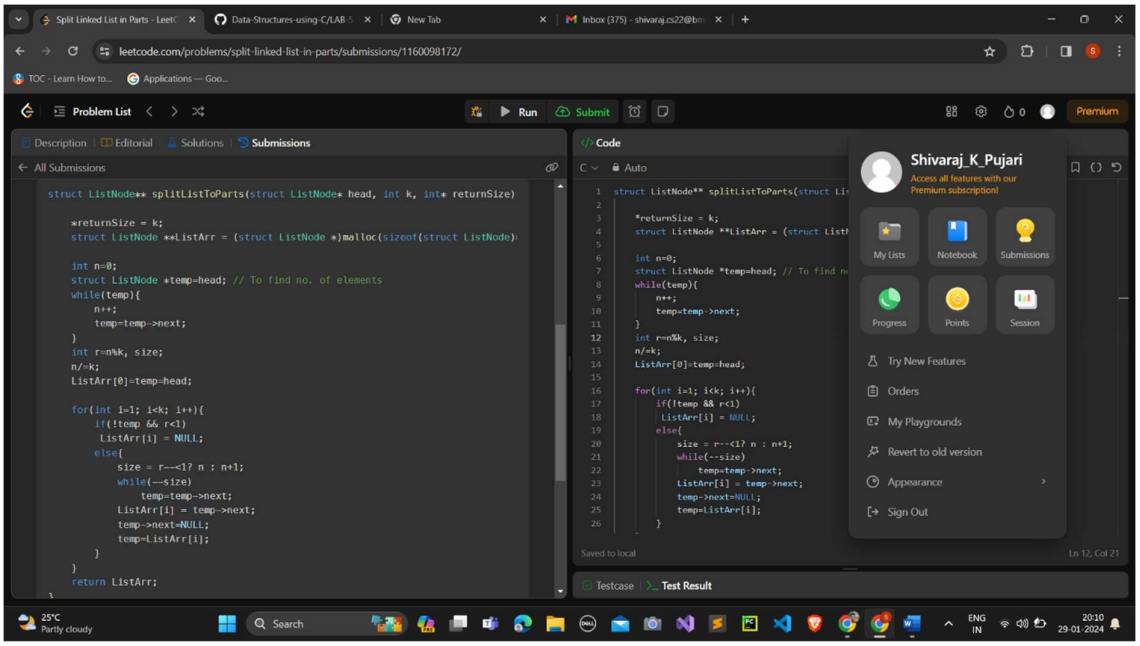
```
        ptr = ptr -> next;
      }
      printf("\n");
   }
}
```

OUTPUT:



LEETCODE PROBLEM: 725. Split Linked List in Parts




Split Linked List in Parts - LeetC
Data-Structures-using-C/LAB-5
New Tab
Inbox (375) - shivaraj.cs22@bm
leetcode.com/problems/split-linked-list-in-parts/submissions/1160098172/
TOC - Learn How to...
Applications — Goo...
Problem List
Run
Submit
Premium
Description
Editorial
Solutions
Submissions
All Submissions
Code
C
Auto
struct ListNode** splitListToParts(struct ListNode* head, int k, int* returnSize)
*returnSize = k;
struct ListNode **ListArr = (struct ListNode *)malloc(sizeof(struct ListNode)
int n=0;
struct ListNode *temp=head; // To find no. of elements
while(temp){
n++;
temp=temp->next;
}
int r=n%k, size;
n/=k;
ListArr[0]=temp=head;
for(int i=1; i<k; i++){
if(!temp && r<1)
ListArr[i] = NULL;
else{
size = r--<1? n : n+1;
while(--size)
temp=temp->next;
ListArr[i] = temp->next;
temp->next=NULL;
temp=ListArr[i];
}
}
return ListArr;
Shivaraj_K_Pujari
Access all features with our Premium subscription!
My Lists
Notebook
Submissions
Progress
Points
Session
Try New Features
Orders
My Playgrounds
Revert to old version
Appearance
Sign Out
Saved to local
Ln 12, Col 21
Testcase
Test Result
25°C
Partly cloudy
Search
ENG
IN
20:10
29-01-2024