structure,

· Easier to implement

· slightly more complex to implement.

· Linear queues are suitable in situations where elements are inserted and removed strictly from one end.

· Circular queues are useful in scenarios where the process of insertion and temoval wraps around.

Tutorials (4/06 /2023

Recap-

1) Why are stacks useful? to call functions and execution, to evaluate expressions, to undo/redo operations to manage memory, to browse history, to backtrack algorithms etc.

@ Reverse a string using stack? # include < stdro.h > # include <string.h>

define MAX_LENGTH 100

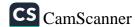
Il function to reverse a String using a stack Void neverse String Cohar " inputstring, char " reveredstring)

> int length = strlen (inputstring); char Stack [MAX_LENG+H]; int top = -1;

```
Il Push each character on to the stack
  for Cint i=0; 1 < length; i++)
    Stack [++top] = input String[i];
   Il Pop pach character from the stack to create the
    reversed string.
int index = 0;
While (top 1 = -1
    reversed String [index ++] = stack [top--];
   perensed String [index ] = sto 1;
int main ()
    char inputString[] = " Hello, world?";
     chas reversed String [MAX-LENGth];
  11 Reverse the string
  reverse String Cinputstring, reversed string):
  11 print the goversed string
  print ("18 n", reversed string;
  11 output " I dirow , ollet "
  geturn o
```

(3) Worte Basic operations of a west. DEnqueue (Insertion) 2) Dequeue (Deletion) 3) Front (Peek) A) Is Empty 5) Is Full What is balanced paranthesis. Balanced parantheses refer to a situation where each opening paranthesis has a corresponding closing parenthesis in the correct order. In other words, for a string containing parentheres, the number of opening parantheres and they should be properly nested. (5) Make stack implementation using a linked 11st - c program. # include a stdro.h> # Include <stlib.h> 11 Structure for a stack Note. struct Node The state of the s Pot data ., Struct Node * next; Il function to create a new node. struct Node * create Note (int data) Commenced Course & State State Course Course struct Node * newNode = (struct Node *) malloc (size of (Struct Node) if (new Node == NULL)

```
Printer ("Memory allocation failed! \n");
                            V system I at
 exit (1):
 newNode -> data = data;
newNode > next = NULL;
 return newNode:
 Il structure for the stack,
Struct Stack
1. Exemple of many of the property
    Struct Node * top .
Il Function to refeate initialize the Stack
Void initialize Stack (struct stack * stack)
 3
      stack -> top = NULL;
Il Function to check if the stack is empty
int is Empty (struct stack * stack)
return stack -> top == NULL:
3
// Function to push and element to the stack
Void push ( struct stack * stuck, int data)
5.01100
struct Node * newNode = create Node (data);
  newNode -> Next = stack -> top:
  stact -> top = newNode:
```



```
and the same of the latter of the same of the
W function to pop an element from the stace
int pop (struct Stack * stack)
 if ( is Empty ( stack))
        pointf ( " stack underflow ! \n " );
        exit CI);
      Struck Node * temp = stack -> top;
 int data = temp -> data ;
 stack - top = temp -> next;
free (temp):
 return data:
// Function to get the top element of the
 inti peek (Struct Stack * stack)
     if (is Propty (stack))
         printf ( et stack is empty! \n ");
        exit (1);
   return stack -> top -> data;
Int main()
struct Stack stack ;
initializeStack (& stack );
 Il puching element onto the
                              stact
         Push ( & stack , 10);
        Push ( & stack , so) ;
           Push ( & Stack , 30);
```

Il Point the top element points (et top element : % d \p" peck (&stack)). A Popping elements from the stack Pernet (ee Popped element : %d \n " . : Pop (& stack)). Prints ("Popped element : 7. d \" , pop (x stack)): If check in the stack is empty. Point ["Is the stack empty? 7-s \n rs Empty (Astack) ? "Yess " "No"). DETALL O . 12023/06/21 Tytorpals (8) (D) What is a execular queue? A circular queux also knows as exprog Buffer " is a data structure that follows first -in- first-out (FIFO) principle. It is implemented as an array / queue with a fixed, size where the last position is connected to the first position / first index forming à circular behaviour. What are the characteristics of circular queue ? pointers in stace 7 Top/perk font and rear as pointers: In circular queues two pointers front maintain the order of the cracular queue. to all equest . The efecular behaviour. : When the dark Structure has reached the capacity the fext