Part A: Stack

Basic level

Write a program in C to work with stacks. Create an example node structure (int value, pointer to the next node) and provide all necessary operations on it:

- insert a new element with a given value to the top of the stack (push)
- remove the element from the top of the stack (pop)
- count the number of elements
- display the stack
- create an example stack
- remove all elements

Advanced level

Write a program in C using stack to check a correct syntaxis for a given expression string x. Examine whether the pairs and the orders of {,},(,),[,] are correct in the given expression.

For example, the function should return 'true' (1) for expression = [()]{}{[()()]()} and 'false' (0) for expression = [(]).

Note:

Working with string you can use arrays of characters, for example: char array[20];

or using dynamic arrays and allocating memory with malloc().

Moreover, you will have to use specific functions of C while doing operations on strings in C, for example:

strlen() - calculates the length of a string

strcpy() - copies a string to another

strcmp() - compares two strings

strcat() - concatenates two strings

Part B: Queue

Basic level

Write a program in C to work with queues. Create an example node structure (int value, pointer to the next node) and provide all necessary operations on it:

- add a new node with a given value at the end of the queue
- remove the first node of the queue and return the value
- count the number of nodes
- display the queue
- create an example queue
- remove all elements

Advanced level

Write a program in C to manage a queue at the web page of ticketmaster to buy tickets for the concert of ColdPlay. There is a fixed number of different ticket types:

General entrance – 200 places – 100€

Tribune seats – 400 places – 120€

Balcony seats – 150 places – 80€

VIP – 50 places – 300€

Gold VIP – 30 places - 500€

Each person that enters the queue wants to buy a given number of tickets of a given type and has a given budget. Simulate the purchase for each person and update the available entrances. Take into consideration:

- If there are no more tickets available of a type that a person wants to buy, just eliminate the person from the queue
- If a person has no sufficient budget for a given ticket type, just eliminate the person from the queue
- If a person can buy a given number of tickets (number of tickets and budget available), process the purchase by updating the number of tickets available after the operation and removing the person from the queue.

Part C: Single linked lists

Basic level

Write a program in C to work with a single linked list. Create an example node structure (int data, pointer to the next node) and provide all necessary operations:

- create and display the list

  Example of an expected output:

  Input data to insert in the list: 5
  Data in the list:
  Data = 5

  Input data to insert in the list: 6

  Data in the list:
  Data = 5
  Data = 6

  Input data to insert in the list: 7

  Data in the list:
  Data = 5
  Data = 6
  Data = 7

- count the number of nodes

  Total number of nodes = 3

- insert a new node at the beginning of the list and display the list

  Input data to insert in the list: 4

  Data in the list:
  Data = 4
  Data = 5
  Data = 6
  Data = 7

- insert a new node at the end of the list and display the list

  Input data to insert in the list: 8

  Data in the list:
  Data = 4
  Data = 5
  Data = 6
  Data = 7
  Data = 8

- insert a new node in the middle of the list and display the list

  Input data to insert in the list: 77

  Input the position after which the new node must be inserted (node data): 7

  Data in the list:

Data = 4
Data = 5
Data = 6
Data = 7
Data = 77
Data = 8

- remove the first node of the list and display the list

  Data in the list:
  Data = 5
  Data = 6
  Data = 7
  Data = 77
  Data = 8

- remove the last node of the list and display the list

  Data in the list:
  Data = 5
  Data = 6
  Data = 7
  Data = 77

- remove a node from the middle of the list and display the list

  Input the position of the node to remove (node data): 7

  Data in the list:
  Data = 5
  Data = 6
  Data = 77

- search an existing element in the list

  Input the element to be searched: 6

  Element found at node number 2

Advanced level

Write a program in C to merge two sorted single linked lists (with integer values) into one sorted single linked list.

Two sorted single linked lists:
1 3 5 7
2 4 6

After merging both sorted lists:
1 2 3 4 5 6 7