In The Name Of God

The Most Compassionate And Merciful

Problem Set -5

Stack

------------------------------------------------

Sheet info :

* Problem Set 5
* Due Date : Not Set
* Just upload scorable questions on HWS.
* File name format : “**studentNumber\_PS5.zip**”
* Do not hesitate to ask any question from your graders!

# 

# 1)write a program to implement stack using array

**properties of stack:(global)**

**Int top\_index**:index of top element of stack

**Int size**:capacity of stack

**Int stack[size]**

You should implement these functions:

# **Void Push(int x):** Adds an item in the stack. If the stack is full print”Stack is FULL.”

# **Int Pop():** Removes an item from the stack. The items are popped in the reversed order in which they are pushed. If the stack is empty print”Stack is EMPTY.”

# **Int Top():** Returns top element of stack If the stack is empty print”Stack is EMPTY.”

# 

# **Int isEmpty():** Returns true if stack is empty, else false.

# 2)write a program to implement queue using array

**properties of queue:(global)**

**Int get\_front()**: returns top element of queue

**Int size**:capacity of queue

**Int queue[size]**

**Int get\_rear():** returns last element of queue

You should implement these functions:

**Void Enqueue(int x):** Adds an item to the queue. If the queue is full, then it is said to be an Overflow condition.

**int Dequeue():** Removes an item from the queue. The items are popped in the same order in which they are pushed. If the queue is empty, then it is said to be an Underflow condition.

**int peek():** function is oftenly used to return the value of first element without dequeuing it.

# 3)write a program to implement queue using 2 stack

# 

# 4)write a program to convert postfix to infix

Input : abc++

Output : (a + (b + c))

Input : ab\*c+

Output : ((a\*b)+c)

# 5)write a program to convert infix to postfix

***Input:*** *str = “a+b\*c-(d/e+f^g^h)”*

***Output:*** *abc\*+de/fgh^^+-*

***Input:*** *a+b\*c*

***Output:*** *abc\*+*

# 6)write a program to convert prefix to infix

Input : Prefix : \*+AB-CD

Output : Infix : ((A+B)\*(C-D))

Input : Prefix : \*-A/BC-/AKL

Output : Infix : ((A-(B/C))\*((A/K)-L))

# 7)write a program to convert infix to prefix

Input : infix :a+b\*(c^d-e)^(f+g\*h)-i

Output : prefix :abcd^e-fgh\*+^\*+i-

# 

# 8)Given a number, find the next smallest palindrome.

|  |
| --- |
| int main() {  int num[] = {2,1,3,3};   int n = 4;   generate\_NextPalindrome( num, n );   return 0; } |

Expected output:

|  |
| --- |
| Next palindrome is:2 2 2 2 |

9)write a simple calculator that can calculate the given

sequence by precedence of operators . the operators can be :

\* , / , + , - , ( , ) , ^

10 ) please write two functions as ​Encrypt​ ​and ​Decrypt

and ​ ​use the in a program with the following menu . ( You

have to use ​Caesar cipher​ .​ if you don’t know

anything about it , google it then )

|  |
| --- |
| < Welcome User > 1 - Encrypt 2 - Decrypt >> 1 Please enter your sentence : >> Hello To C students . Please Enter the Key : >>1285 The Encrypted Sentence is : m#\*\*!>C!>^>sC['#vCs |

11)assume we have an array of integers , write a function that

finds the subarray with maximum summation of elements.

12)Get scientific number

Using only the getchar () function for getting Input, then Write a

function that gets a number in scientific notation and returns it

as a double and print it using putchar function.

Prototype:

|  |
| --- |
| double next\_sc (char num[]); |

Examples:

|  |
| --- |
| next\_sc ("12e-1") --> next\_sc ("12e9") --> next\_sc ("0e+999") --> returns: 1.2 returns: 12000000000  returns: 0 |

13) write a program that evaluates the following expression

display the result in integer format.

ans = 7 times 9 plus ( 19 divided by 5 ) modulo 2

int ans = 7 \* 9 + ( 19 / 5 ) % 2

14) write a function that gets multiple lines of input from

the user and then converts the “newline” , “tab” , “space”

into visible escape sequences such as “\n” , “\t” , “b”.you

have to use switch case . write a function for other

direction as well ( for example converts “\t” to tab ).

15)Write a program to get a positive number n as an input. And print nth prime number.

Input Number n written in a single line and it is lower than 3×〖10〗^5. Maximum 20 problems will

be given and every problem should answer less than 0.5 seconds.

Output Write nth prime number

Input :

2

Output:

3

Input :

3

Output:

5

Input:

299999

Output:

4256227

# 16)A queue is a data structure based on the principle of 'First In First Out' (FIFO). There are two ends; one end can be used only to insert an item and the other end to remove an item. A Double Ended Queue is a queue where you can insert an item in both sides as well as you can delete an item from either side. There are mainly four operations available to a double ended queue. They are:

# 1. **pushLeft():** inserts an item to the left end of the queue with the exception that the queue is not full.

# 2. **pushRight():** inserts an item to the right end of the queue with the exception that the queue is not full.

# 3. **popLeft():** removes an item from the left end of the queue with the exception that the queue is not empty.

# 4. **popRight():** removes an item from the right end of the queue with the exception that the queue is not empty.

# 

# Implement double ended queue using array.

18)A string is called *diverse* if it contains consecutive (adjacent) letters of the Latin alphabet and each letter occurs exactly once. For example, the following strings are diverse: "fced", "xyz", "r" and "dabcef". The following string are **not** diverse: "az", "aa", "bad" and "babc". Note that the letters 'a' and 'z' are not adjacent.

Formally, consider positions of all letters in the string in the alphabet. These positions should form contiguous segment, i.e. they should come one by one without any gaps. And all letters in the string should be distinct (duplicates are not allowed).

You are given a sequence of strings. For each string, if it is diverse, print "Yes". Otherwise, print "No".

**Input**

The first line contains integer n

n lines contains strings, one string per line. Each string contains only lowercase Latin letters, its length is between 1 and 100

**Output**

Print n lines, one line per a string in the input. The line should contain "Yes" if the corresponding string is diverse and "No" if the corresponding string is not diverse. You can print each letter in any case (upper or lower). For example, "YeS", "no" and "yES" are all acceptable.

# 

**Example**

**input**

8

fced

xyz

r

dabcef

az

aa

bad

babc

**output**

Yes

Yes

Yes

Yes

No

No

No

No