

2. Algorithm

1. SI set $top \rightarrow 0$
2. Declare an array a of size n
3. Define a function push
 - 3.a. Check if $top \rightarrow n-1$
 - 3.a.1 Display stack overflow
 - 3.b. if top not equal to $n-1$
 - 3.b.1 increment top by 1
 - 3.b.2 set array of $top \rightarrow$ element occupied
4. Define a function pop
 - 4.a. define a variable b
 - 4.b. Check if $top \rightarrow -1$
 - 4.b.1 Display stack underflow
 - 4.c. if not $top \rightarrow -1$
 - 4.c.1 $b \rightarrow$ array(top)
 - 4.c.2 decrement top by 1
 - 4.c.3 display b
5. Define a for display
 - 5.a. generate a loop from $i \rightarrow top$ to $i \rightarrow 0$
 - 5.a.1 Display add(top)
 - 5.b. decrement i by 1

Pseudo code

Pseudo code

int top = -1, stack[Max];

push(x)

Scanned with CamScanner

classmate

Date

Page

top = top + 1

stack[top] = x

}

pop()

{

top = top - 1

}

IsEmpty()

{

if (top == -1)

return true;

else

return false;

IsFull()

{

if (top == size)

return true;

else

return false;

display()

{

for (i = top; i >= 0; --i)

{

printf("%d\n", stack[i]);

}

}

Scanned with CamScanner