Practical 4

Name: Shantanu Sethi

Roll no. 163

Aim: To implement different operation on stack using array

1. . P u s h ( )
2. . P o p ( )

3.i s f u l l ( )

4 . i s e m p t y ( )

1. c o u n t ( )
2. display()

Objectives:

1 . L e a r n h o w t o i m p l e m e n t d i f f e r e n t o p e r a t i o n o n s t a c k u s i n g a r

Theory:

A stack is an ordered collection of items into which new items may be inserted and items may be deleted at one end called TOP of the stack.

A stack is a homogeneous collection of items of any one type. Data can be added or removed from only the top.

Last In, First Out (LIFO)

Operations perform on stack:

Push : Place an item onto the stack. If there is no place for new item, stack is in overflow state.

Algorithm:

Algorithm :push( S, TOP, X ): This algorithm insert element x to the top of the stack which is represented by array S containing N elements with pointer TOP denoting the top most element in the stack.

* 1. [ check for stack overflow]

if TOP >= N-1

write[ stack overflow] return

* 1. [ Increment TOP ]

TOP = TOP+1

* 1. [ Insert element ]

S[TOP]=X

* 1. [finished] return

Pop :Return the item at the top of the stack and then remove it. If pop is called when stack is empty, it is in an underflow state.

Algorithm :pop( S, TOP ): This algorithm remove top most element from top of the stack which is represented by array S containing N elements with pointer TOP denoting the top most element in the stack. 1. [ check for stack underflow] if TOP = -1

write[ stack underflow on POP] return

1. [ Decrement TOP Pointer ] TOP = TOP-1
2. [ return top element from stack ] return(S[TOP+1]

isfull: Tells if the stack is full or not.

Algorithm :isfull( S, TOP, N ): This algorithm check whether stack is full . stack S containing N elements with pointer TOP denoting the top most element in the stack.

1. [ check for stack overflow] if TOP > =N-1

write[ stack full] return

isEmpty : Tells if the stack is empty or not.

Algorithm :isempty l( S, TOP, N ): This algorithm check whether stack is full .

stack S containing N elements with pointer TOP denoting the top most element in the stack.

1. [ check for stack overflow] if TOP ==-1

write[ stack empty]

return

Count:The number of items in the stack.

Algorithm :count( S, TOP ):This algorithm count number of elements present in stack which is represented by array S containing N elements with pointer TOP denoting the top most element in the stack.

1. [check for empty stack ] if TOP==-1 count=0 return count
2. For i=0 to top Count=count+1
3. Return count

Change():

Algorithm :change( S, TOP,X i ):This algorithm display elements present in stack from top of the stack S containing N elements with pointer TOP denoting the top most element in the stack.

s

* 1. [check for empty stack ] if TOP==-1

display(“stack is empty”)

* 1. For i=top to 0

Display ( s[i])

* 1. [ finish ] return

1. Push() CODE:



p a c k a g e s t a c k ; p u b l i c c l a s s p u s h { p u b l i c s t a t i c v o i d m a i n ( S t r i n g [ ] a r g s ) {

/ / T O D O A u t o - g e n e r a t e d m e t h o d s t u b

S y s t e m . o u t . p r i n t l n ( " 1 8 4 - A b h i n a v S i n g h " ) ; i n t m a x S i z e = 5 ; / / M a x i m u m s i z e o f t h e s t a c k i n t [ ] S = n e w i n t [ m a x S i z e ] ; / / A r r a y t o s t o r e s t a c k e l e m e n t i n t T O P = - 1 ; / / P o i n t e r t o t h e t o p o f t h e s t a c k

/ / P u s h e l e m e n t s o n t o t h e s t a c k

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k o v e r f l o w " ) ;

} e l s e {

S [ + + T O P ] = 1 0 ; / / P u s h 1 0

}

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k o v e r f l o w " ) ;

} e l s e {

S [ + + T O P ] = 2 0 ; / / P u s h 2 0

}

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k o v e r f l o w " ) ;

} e l s e {

S [ + + T O P ] = 3 0 ; / / P u s h 3 0

}

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k o v e r f l o w " ) ;

} e l s e {

S [ + + T O P ] = 4 0 ; / / P u s h 4 0

}

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k o v e r f l o w " ) ;

} e l s e {

S [ + + T O P ] = 5 0 ; / / P u s h 5 0

}

/ / A t t e m p t t o p u s h o n e m o r e e l e m e n t t o t e s t o v e r f l o w

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k o v e r f l o w " ) ;

} e l s e {

S [ + + T O P ] = 6 0 ; / / T h i s s h o u l d c a u s e o v e r f l o w

}

/ / D i s p l a y t h e s t a c k e l e m e n t s

S y s t e m . o u t . p r i n t l n ( " S t a c k e l e m e n t s : " ) ;

f o r ( i n t i = T O P ; i > = 0 ; i - - ) {

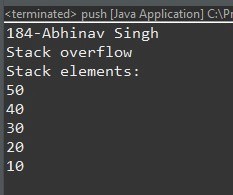
S y s t e m . o u t . p r i n t l n ( S [ i ] ) ;

}

}

}

OUTPUT:-



# 2. Pop()

CODE:-

p a c k a g e s t a c k ; p u b l i c c l a s s p o p {

p u b l i c s t a t i c v o i d m a i n ( S t r i n g [ ] a r g s ) {

/ / T O D O A u t o - g e n e r a t e d m e t h o d s t u b

S y s t e m . o u t . p r i n t l n ( " 1 8 4 - A b h i n a v S i n g h " ) ;

i n t m a x S i z e = 5 ; / / M a x i m u m s i z e o f t h e s t a c k

i n t [ ] S = n e w i n t [ m a x S i z e ] ; / / A r r a y t o s t o r e s t a c k e l e m e n t s

i n t T O P = - 1 ; / / P o i n t e r t o t h e t o p o f t h e s t a c k

/ / P u s h e l e m e n t s o n t o t h e s t a c k

i f ( T O P < m a x S i z e - 1 ) {

S [ + + T O P ] = 1 0 ; / / P u s h 1 0

}

i f ( T O P < m a x S i z e - 1 ) {

S [ + + T O P ] = 2 0 ; / / P u s h 2 0

}

i f ( T O P < m a x S i z e - 1 ) {

S [ + + T O P ] = 3 0 ; / / P u s h 3 0

}

/ / D i s p l a y t h e s t a c k e l e m e n t s

S y s t e m . o u t . p r i n t l n ( " S t a c k e l e m e n t s : " ) ;

f o r ( i n t i = T O P ; i > = 0 ; i - - ) {

S y s t e m . o u t . p r i n t l n ( S [ i ] ) ;

}

/ / P o p a n e l e m e n t f r o m t h e s t a c k

i f ( T O P = = - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k u n d e r f l o w o n P O P " ) ;

} e l s e { i n t p o p p e d E l e m e n t = S [ T O P - - ] ; / / P o p t h e t o p e l e m e n t S y s t e m . o u t . p r i n t l n ( " P o p p e d e l e m e n t : " + p o p p e d E l e m e n t ) ;

}

/ / D i s p l a y t h e s t a c k e l e m e n t s a f t e r p o p

S y s t e m . o u t . p r i n t l n ( " S t a c k e l e m e n t s a f t e r p o p : " ) ;

f o r ( i n t i = T O P ; i > = 0 ; i - - ) {

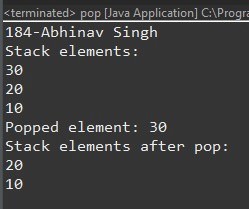
S y s t e m . o u t . p r i n t l n ( S [ i ] ) ;

}

}

}

## OUTPUT:-



3. isfull()

## CODE:-

p a c k a g e s t a c k ; p u b l i c c l a s s i s f u l l { p u b l i c s t a t i c v o i d m a i n ( S t r i n g [ ] a r g s ) {

/ / T O D O A u t o - g e n e r a t e d m e t h o d s t u b S y s t e m . o u t . p r i n t l n ( " 1 8 4 - A b h i n a v S i n g h " ) ; i n t m a x S i z e = 5 ; / / M a x i m u m s i z e o f t h e s t a c k

i n t [ ] S = n e w i n t [ m a x S i z e ] ; / / A r r a y t o s t o r e s t a c k e l e m e n t s

i n t T O P = - 1 ; / / P o i n t e r t o t h e t o p o f t h e s t a c k

/ / P u s h e l e m e n t s o n t o t h e s t a c k

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k f u l l " ) ;

} e l s e {

S [ + + T O P ] = 1 0 ; / / P u s h 1 0

}

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k f u l l " ) ;

} e l s e {

S [ + + T O P ] = 2 0 ; / / P u s h 2 0

}

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k f u l l " ) ;

} e l s e {

S [ + + T O P ] = 3 0 ; / / P u s h 3 0

}

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k f u l l " ) ;

} e l s e {

S [ + + T O P ] = 4 0 ; / / P u s h 4 0

}

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k f u l l " ) ;

} e l s e {

S [ + + T O P ] = 5 0 ; / / P u s h 5 0

}

/ / A t t e m p t t o p u s h o n e m o r e e l e m e n t t o t e s t o v e r f l o w

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k f u l l " ) ;

} e l s e {

S [ + + T O P ] = 6 0 ; / / T h i s s h o u l d n o t b e a d d e d

}

/ / D i s p l a y t h e s t a c k e l e m e n t s

S y s t e m . o u t . p r i n t l n ( " S t a c k e l e m e n t s : " ) ; f o r ( i n t i = T O P ; i > = 0 ; i - - ) {

S y s t e m . o u t . p r i n t l n ( S [ i ] ) ;

}

/ / C h e c k i f t h e s t a c k i s f u l l

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k i s f u l l " ) ;

} e l s e {

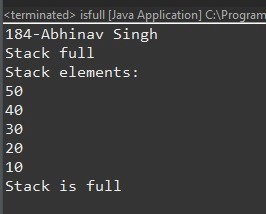
S y s t e m . o u t . p r i n t l n ( " S t a c k i s n o t f u l l " ) ;

}

}

}

## OUTPUT:-



# 4. isempty()

CODE:-

p a c k a g e s t a c k ; p u b l i c c l a s s i s e m p t y {

p u b l i c s t a t i c v o i d m a i n ( S t r i n g [ ] a r g s ) {

S y s t e m . o u t . p r i n t l n ( " 1 8 4 - A b h i n a v S i n g h " ) ; i n t m a x S i z e = 5 ; / / M a x i m u m s i z e o f t h e s t a c k

i n t [ ] S = n e w i n t [ m a x S i z e ] ; / / A r r a y t o s t o r e s t a c k e l e m e n t s i n t T O P = - 1 ; / / P o i n t e r t o t h e t o p o f t h e s t a c k

/ / C h e c k i f t h e s t a c k i s e m p t y i f ( T O P = = - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k e m p t y " ) ;

} e l s e {

S y s t e m . o u t . p r i n t l n ( " S t a c k i s n o t e m p t y " ) ;

}

/ / P u s h e l e m e n t s o n t o t h e s t a c k

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k f u l l " ) ;

} e l s e {

S [ + + T O P ] = 1 0 ; / / P u s h 1 0

}

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k f u l l " ) ;

} e l s e {

S [ + + T O P ] = 2 0 ; / / P u s h 2 0

}

/ / C h e c k i f t h e s t a c k i s e m p t y a g a i n i f ( T O P = = - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k e m p t y " ) ;

} e l s e {

S y s t e m . o u t . p r i n t l n ( " S t a c k i s n o t e m p t y " ) ;

}

/ / D i s p l a y t h e s t a c k e l e m e n t s

S y s t e m . o u t . p r i n t l n ( " S t a c k e l e m e n t s : " ) ;

f o r ( i n t i = T O P ; i > = 0 ; i - - ) {

S y s t e m . o u t . p r i n t l n ( S [ i ] ) ;

}

/ / P o p e l e m e n t s f r o m t h e s t a c k

i f ( T O P = = - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k e m p t y " ) ;

} e l s e { i n t p o p p e d E l e m e n t = S [ T O P - - ] ; / / P o p t h e t o p e l e m e n t S y s t e m . o u t . p r i n t l n ( " P o p p e d e l e m e n t : " + p o p p e d E l e m e n t ) ;

}

/ / F i n a l c h e c k i f t h e s t a c k i s e m p t y i f ( T O P = = - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k e m p t y " ) ;

} e l s e {

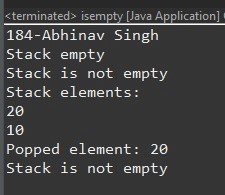
S y s t e m . o u t . p r i n t l n ( " S t a c k i s n o t e m p t y " ) ;

}

}

}

## OUTPUT:-



# 5. count() CODE:-

p a c k a g e s t a c k ; p u b l i c c l a s s c o u n t {

p u b l i c s t a t i c v o i d m a i n ( S t r i n g [ ] a r g s ) { S y s t e m . o u t . p r i n t l n ( " 1 8 4 - A b h i n a v S i n g h " ) ; i n t m a x S i z e = 5 ; / / M a x i m u m s i z e o f t h e s t a c k

i n t [ ] S = n e w i n t [ m a x S i z e ] ; / / A r r a y t o s t o r e s t a c k e l e m e n t s i n t T O P = - 1 ; / / P o i n t e r t o t h e t o p o f t h e s t a c k

/ / P u s h e l e m e n t s o n t o t h e s t a c k

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k f u l l " ) ;

} e l s e {

S [ + + T O P ] = 1 0 ; / / P u s h 1 0

}

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k f u l l " ) ;

} e l s e {

S [ + + T O P ] = 2 0 ; / / P u s h 2 0

}

i f ( T O P > = m a x S i z e - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k f u l l " ) ;

} e l s e {

S [ + + T O P ] = 3 0 ; / / P u s h 3 0

}

/ / C o u n t t h e n u m b e r o f i t e m s i n t h e s t a c k i n t c o u n t ; i f ( T O P = = - 1 ) {

c o u n t = 0 ; / / S t a c k i s e m p t y

} e l s e { c o u n t = 0 ; / / I n i t i a l i z e c o u n t f o r ( i n t i = 0 ; i < = T O P ; i + + ) {

c o u n t + + ; / / I n c r e m e n t c o u n t f o r e a c h e l e m e n t

}

}

/ / D i s p l a y t h e c o u n t o f e l e m e n t s i n t h e s t a c k

S y s t e m . o u t . p r i n t l n ( " N u m b e r o f i t e m s i n t h e s t a c k : " + c o u n t ) ;

/ / P o p e l e m e n t s f r o m t h e s t a c k

i f ( T O P = = - 1 ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k e m p t y " ) ;

} e l s e { i n t p o p p e d E l e m e n t = S [ T O P - - ] ; / / P o p t h e t o p e l e m e n t

S y s t e m . o u t . p r i n t l n ( " P o p p e d e l e m e n t : " + p o p p e d E l e m e n t ) ;

}

/ / F i n a l c o u n t a f t e r p o p p i n g i f ( T O P = = - 1 ) {

c o u n t = 0 ; / / S t a c k i s e m p t y

} e l s e { c o u n t = 0 ; / / I n i t i a l i z e c o u n t f o r ( i n t i = 0 ; i < = T O P ; i + + ) {

c o u n t + + ; / / I n c r e m e n t c o u n t f o r e a c h e l e m e n t

}

}

/ / D i s p l a y t h e c o u n t o f e l e m e n t s i n t h e s t a c k a f t e r p o p p i n g

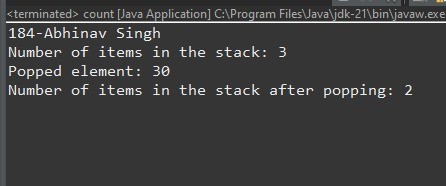
S y s t e m . o u t . p r i n t l n ( " N u m b e r o f i t e m s i n t h e s t a c k a f t e r p o p p i n g :

/ / T O D O A u t o - g e n e r a t e d m e t h o d s t u b

}

}

## OUTPUT:-



6. display() CODE:-

p a c k a g e s t a c k ; i m p o r t j a v a . u t i l . S t a c k ; p u b l i c c l a s s d i s p l a y {

p u b l i c s t a t i c v o i d c h a n g e ( S t a c k < I n t e g e r > s t a c k ) { i f ( s t a c k . i s E m p t y ( ) ) {

S y s t e m . o u t . p r i n t l n ( " S t a c k i s e m p t y " ) ; r e t u r n ;

}

f o r ( i n t i = s t a c k . s i z e ( ) - 1 ; i > = 0 ; i - - ) {

S y s t e m . o u t . p r i n t l n ( s t a c k . g e t ( i ) ) ;

}

}

p u b l i c s t a t i c v o i d m a i n ( S t r i n g [ ] a r g s ) {

S y s t e m . o u t . p r i n t l n ( " 1 8 4 - A b h i n a v S i n g h " ) ; S t a c k < I n t e g e r > s t a c k = n e w S t a c k < > ( ) ;

s t a c k . p u s h ( 1 0 ) ; s t a c k . p u s

h ( 2 0 ) ;

sc thaacnkg.ep(usst a c k ) ;

} h ( 3 0 ) ;

}

## OUTPUT:-

