DSA Lab

Name: Siddhi Parekh

Reg No.: 221071047

Batch: C

SY Comps

Experiment No.: 2

AIM:

To add very large integer numbers using the stack data structure.

THEORY:

Long Integer Arithmetic Addition involves adding two integers that may exceed the capacity of standard integer data types. To perform this operation, a stack-based approach can be employed. The theory below outlines the key concepts and steps involved in implementing long integer addition using a stack.

1. Data Representation:
   - The long integers are represented as integer arrays

2. Stack Implementation:
   - A stack data structure is used to facilitate the addition operation. The stack allows us to process the digits from right to left, mimicking the manual addition process.

ALGORITHM:

Step 0: Start

Step 1: Define functions of stack class like push, pop, empty, full and peek.

Step 2: Inside the main function take the input for 2 large integers, upto 100, by making 2 objects of class stack
Step 3: push the integers when taken input into the stack, by unit place number at the top of the stack.
Step 4: pop the digits till stack is empty, add them, and consider the carry and remainder. Push the remainder in the third object of the class.
Step 5: pop the elements of 3rd stack to get the final output.
Step 6: Stop

EXAMPLE:

let's take 2 numbers: 89 & 13

->89 is put on stack 1 in order as 8 first and then 9
and 13 is put on stack 2 in order as 1 first and then 3

-> 9 from stack1 and 3 from stack 2 is popped and added to it gives 12.
12%10 i.e. 2 is pushed onto stack 3
and carry equals 12/10 i.e 1

->now 8 and 1 is popped off stack 1 and stack 2 and added to carry which is 1
sum= 8 +1 +1 =10
10%10 =1 is pushed onto stack 3 and carry =10/10=1 which is not 0 and both stacks 1 and 2 are empty, hence it is pushed onto stack 3.

->now ans is pushed in the stack in the order

->now while printing on the screen we pop elements of stack 3 while it is not empty and print it on the screen. hence on the screen it is displayed as 102 .

CONCLUSION:

Thus, from this experiment we have gained knowledge of stacks and successfully added two large numbers using stacks.

CODE:

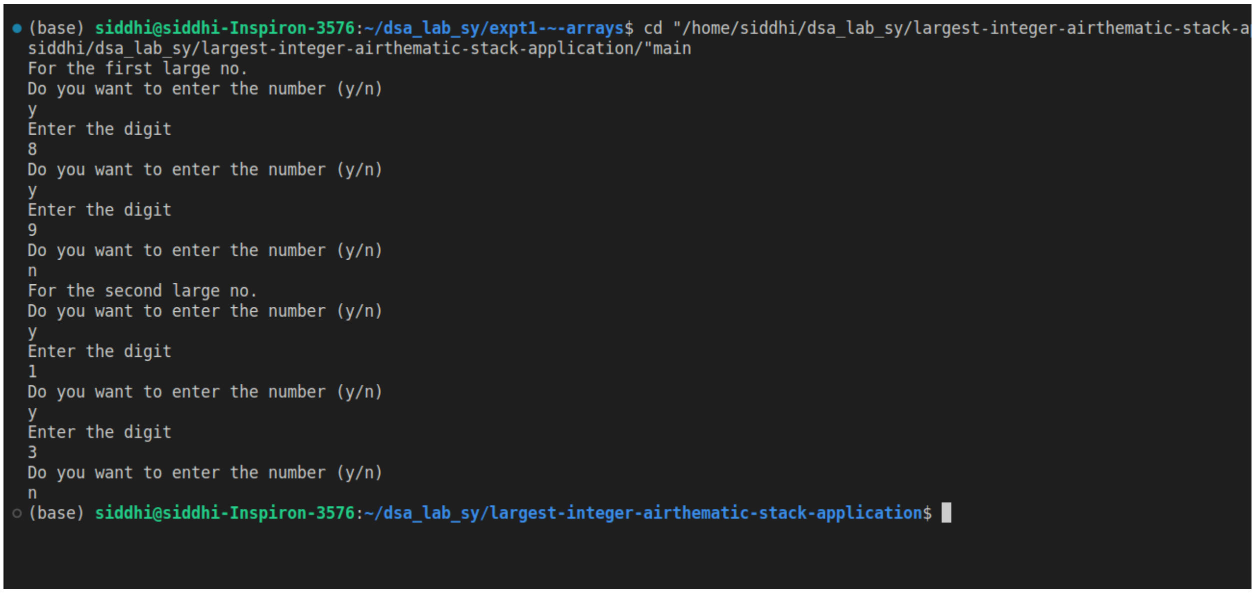
```
#include<iostream>
using namespace std;

class stk
{
  int top;
  int arr1[100];

  public :

  stk()
  {
        top=-1;
  }

  bool isfull()
  {
        if(top==99)
        {
        return 1;
        }

        else
        {
        return 0;
        }
  }

  bool isempty()
  {
        if(top==-1)
        {
        return 1;
        }
```

```
	else
	{
	return 0;
	}
}

bool cond1()
{
	char a='y';
	char b='n';
	char c;

	cout<<"Do you want to enter the number (y/n)"<<endl;
	cin>>c;

	if(c==a)
	{
	cout<<"Enter the digit"<<endl;
	return 1;
	}

	else if(c==b)
	{
	//break;						//cannot use here
	return 0;
	}

	else
	{
	cout<<"ERROR: INVALID INPUT"<<endl;
	return 0;
	}
}

void push(int data)
{
        if(isfull())
        {
        cout<<"Stack Overflow"<<endl;
        }

        else
        {
        top = top+1;
        arr1[top]=data;
        }
}

 int pop()
 {
        if(isempty())
        {
        cout<<"Stock Underflow"<<endl;
        return 0;
        }

        else
        {
        return arr1[top--];
        }
 }

 int peek()
 {
        if(isempty())
        {
        cout<<"Stack Underflow"<<endl;
        return 0;
        }

        else
        {
        return arr1[top];
        }

   }
};

int main()
{
        int d;
        stk s,s1,s2;

        cout<<"For the first large no."<<endl;
        while(s.cond1())
        {
        cin>>d;
        s.push(d);
        }

        cout<<"For the second large no."<<endl;
        while(s1.cond1())
        {
        cin>>d;
        s1.push(d);
        }


        int e,f,g,q,r,x;
        int carry=0;

        while((!s1.isempty()) && (!s.isempty()))
        {
        f=s.pop();
        g=s1.pop();
        e=f+g+carry;
        if(e>=10)
        {
        r=e%10;
        q=e/10;
```

```
carry=q;
s2.push(r);
}
else{
s2.push(e);}

}

while(!s1.isempty())
{
f=s.pop();
//g=s1.pop();
e=f+carry;
if(e>=10)
{
r=e%10;
q=e/10;
carry=q;
s2.push(r);
}
else{
s2.push(e);}
}

while(!s.isempty())
{
//f=s.pop();
g=s1.pop();
e=g+carry;
if(e>=10)
{
r=e%10;
q=e/10;
carry=q;
s2.push(r);
}
else{
```

```
          s2.push(e);}
          }

          if(carry==1)
          {
          s2.push(1);
          }

          cout<<"The sum of 2 large no.s entered is: ";

          while(!s2.isempty())
          {
          cout<<s2.pop();
          }



}
```

OUTPUT:



```
● (base) siddhi@siddhi-Inspiron-3576:~/dsa_lab_sy/expt1-~-arrays$ cd "/home/siddhi/dsa_lab_sy/largest-integer-airthematic-stack-a
  siddhi/dsa_lab_sy/largest-integer-airthematic-stack-application/"main
  For the first large no.
  Do you want to enter the number (y/n)
  y
  Enter the digit
  8
  Do you want to enter the number (y/n)
  y
  Enter the digit
  9
  Do you want to enter the number (y/n)
  n
  For the second large no.
  Do you want to enter the number (y/n)
  y
  Enter the digit
  1
  Do you want to enter the number (y/n)
  y
  Enter the digit
  3
  Do you want to enter the number (y/n)
  n
○ (base) siddhi@siddhi-Inspiron-3576:~/dsa_lab_sy/largest-integer-airthematic-stack-application$ ▌
```