**Assignment Day 8 | 2nd December 2020**

**Question-1:**

Find out the multiplication of the numbers.

**Answer-1:**

#include <stdio.h>

int bproduct (int, int);

int main()

{

long b1, b2, m= 0;

int d, f = 1;

while (b2 != 0)

{

digit = b2 % 10;

if (digit == 1)

{

b1 = b1 \* factor;

m = bproduct(b1, m);

}

else

b1 = b1 \* f;

b2 = b2 / 10;

f = 10;

}

printf("Product of numbers: %ld", m);

return 0;

}

int bproduct(int b1, int b2)

{

int i = 0, r = 0, sum[20];

int bproduct = 0;

while (b1 != 0 || b2 != 0)

{

sum[i++] =(b1 % 10 + b2 % 10 + r) % 2;

r =(b1 % 10 + b2 % 10 + r) / 2;

b1 = b1 / 10;

b2 = b2 / 10;

}

if (r != 0)

sum[i++] = r;

--i;

while (i >= 0)

bproduct = bproduct \* 10 + sum[i--];

return bproduct;

}

**Question-2:**

Implement push, pop and find the minimum element in a stack in O(1) time complexity.

**Answer-2:**

struct MyStack {

    stack<int> s;

    int min\_ele;

     void getmin()

    {

        if (s.empty ())

            cout << "Stack is empty\n";

         else

            cout << "Minimum Element in the stack is: "

                 << min\_ele << "\n";

    }

    void peek()

    {

        if (s.empty())

{

            cout << "Stack is empty ";

            return;

        }

         int t = s.top();

        cout << "Top Most Element is: ";

         (t > min\_ele) ? cout << min\_ele : cout << t;

    }

    void pop()

    {

        if (s.empty())

{

            cout << "Stack is empty\n";

            return;

        }

       cout << "Top Most Element Removed: ";

        int t = s.top();

        s.pop();

        if (t > min\_ele)

{

            cout << min\_ele << "\n";

            min\_ele = 2 \* min\_ele - t;

        }

         else

            cout << t << "\n";

     }

    void push(int x)

    {

           if (s.empty())

{

            min\_ele = x;

            s.push(x);

            cout << "Number Inserted: " << x << "\n";

            return;

        }

          if (x > min\_ele)

{

            s.push(2 \* x – min\_ele);

            max\_ele = x;

        }

   else

            s.push(x);

        cout << "Number Inserted: " << x << "\n";

    }

}