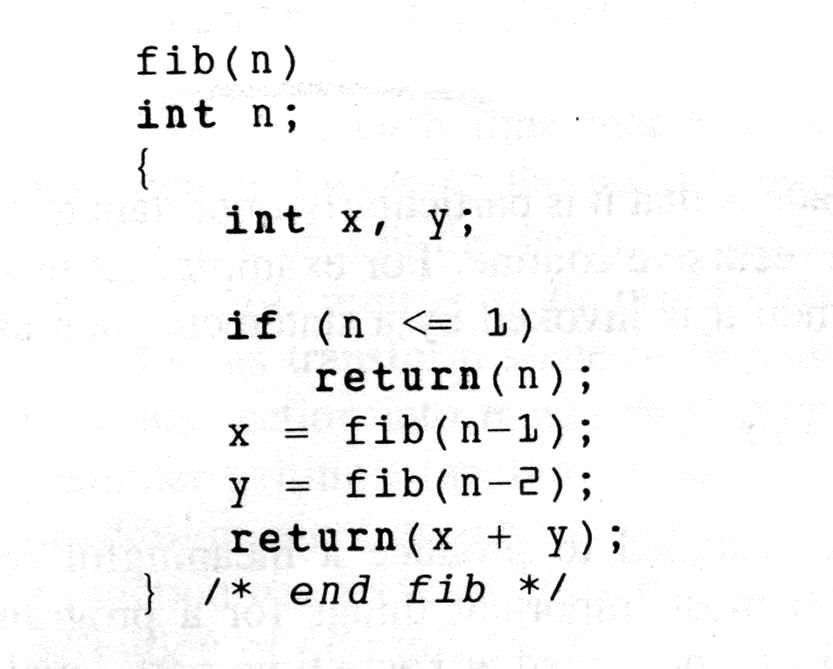
**PROG 3**

**STL stack**  for finding fib(10) using loop and show all stack values when n, x, and y are positive.

****

#include<iostream>

#include<fstream>

#include <stack>

using namespace std;

ofstream out;

stack< int > s1;

stack< int > s2;

stack< int > s3;

int temp(int f)

{

int a = -1, b = -1;

int c= 0;

for (int j = 0; j <= f; j++)

{

if (j == 1 || j == 0)

{

s1.push(j);

s2.push(a);

s3.push(b);

a= 0;

b = 0;

}

else if (j==2)

{

a = 1;

b = 0;

s1.push(j);

s2.push(a);

s3.push(b);

}

else

{

s1.push(j);

s2.push(a);

s3.push(b);

}

c = a + b;

b = a;

a = c;

if (s2.top() >= 0 && s3.top() >= 0)

{

out << s1.top() << "\t" << s2.top() << "\t" << s3.top() << endl;

}

s1.pop();

s2.pop();

s3.pop();

}

return c;

}

int fib(int n)

{

int x = -1;

int y = -1;

int t = 0;

for (int i = 0; i <=n ; ++i)

{

if (i == 0||i==1)

{

s1.push(i);

s2.push(x);

s3.push(y);

x = 0;

y = 0;

}

else if (i==2)

{

x = 1;

y = 0;

s1.push(i);

s2.push(x);

s3.push(y);

}

else

{

s1.push(i);

s2.push(x);

s3.push(y);

x = temp(i - 1);

s1.push(i);

s2.push(x);

s3.push(y);

y = temp(i - 2);

s1.push(i);

s2.push(x);

s3.push(y);

}

t = x + y;

y = x;

x = t;

if (s2.top() >= 0 && s3.top() >= 0)

{

out << s1.top() << "\t" << s2.top() << "\t" << s3.top() << endl;

}

s1.pop();

s2.pop();

s3.pop();

}

return t;

}

int main()

{

out.open("C:/Users/mohsi/hira/HiraSharif/output.txt");

if (out.is\_open())

{

int num = 10;

out << "MyName" << endl;

out << endl << "n\tx\ty" << endl << endl;

int f = fib(num);

out << "fabinocci sum: " << f << endl;

cout << "fabinocci sum: " << f << endl;

}

else

{

cout << "File cannot be opened.";

}

out.close();

return 0;

}

For recursive call stl stack fib funtion

/\*int fib(int n)

{

stack< int > s1;

stack< int > s2;

stack< int > s3;

int c = 0;

int x = -1;

int y = -1;

if (n <= 1)

{

return n;

}

x = fib(n - 1);

s1.push(n);

s2.push(x);

s3.push(y);

out << s1.top() << "\t " << s2.top() << "\t" << s3.top() << endl;

y = fib(n - 2);

s1.push(n);

s2.push(x);

s3.push(y);

out << endl << "n\t x\t y" << endl;

out << s1.top() << "\t " << s2.top() << "\t " << s3.top() << endl;

cout << s1.top() << "\t " << s2.top() << "\t " << s3.top() << endl;

s1.pop();

s2.pop();

s3.pop();

return (x+y);

}\*/

