1. For following example, use a sequence of diagrams like those in the text (Section 7.4) to trace the action of the run-time stack.

```
#include <iostream>
void fl(int );
int f2(int p);
int f3(int n);


int main()
{
int a = 5;
f1(a);
cout << endl; // A: Return here after f1 finishes
}

//--- Function fl
void fl(int x)
{
cout << f2(x + 1); // B: Return here after f2 finishes
}

//--- Function f2
int f2(int p)
{
int q = f3(p/3); // C: Return here after f3 finishes
return 3*q + 1;
}

/--- Function f3
int f3(int n)
{
return n*n + 1;
}
```

	Parameters	Function Value	Local Variables	Return Address
Activation Record			a 5	A

Function call f1(a)

	Parameters	Function Value	Local Variables	Return Address
AR for f1()	x 5			A
AR for main()			a 5	OS

Function call for f2(x+1)

	Parameters	Function Value	Local Variables	Return Address
AR for f2()	p 6		q	B
AR for f1()	x 5			A
AR for main()			a 5	OS

Function call f3(p/3)

	Parameters	Function Value	Local Variables	Return Address
AR for f3()	n 2			C
AR for f2()	p 6		q	B
AR for f3()	x 5			A
AR for main()			a 5	OS

Return from f(3)

	Parameters	Function Value	Local Variables	Return Address
POP AR for f3()	n 2	**5**		C

	Parameters	Function Value	Local Variables	Return Address
AR for f2()	p 6		q	B
AR for f1()	x 5			A
AR for main()			a 5	OS

Return from f2()

POP AR for f2()	p 6	16	q 5	B

AR for f1()	x 5			A
AR for main()			a 5	OS

Return from f1()

POP AR for f1()	x 5			A

AR for main()			a 5	OS

Return from main()

POP AR form main()			a 5	OS