

Stack

Call Method with
Value Types Arguments

```
static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

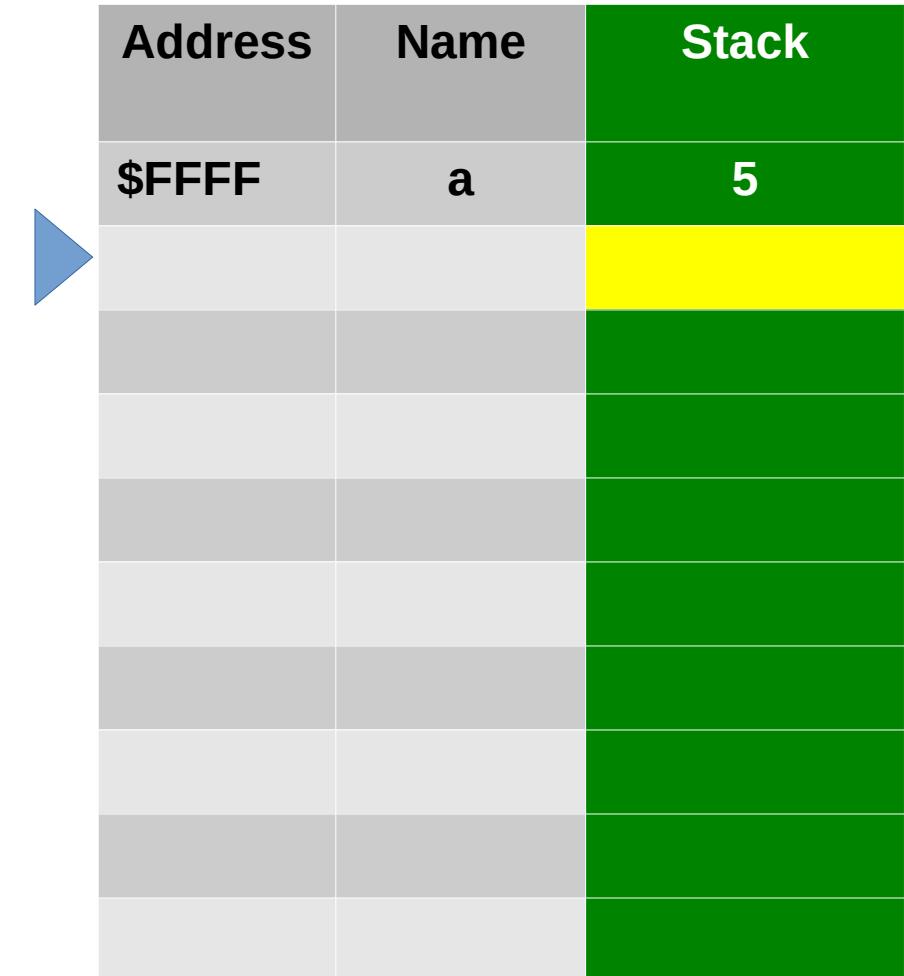
    return res;
}
```



```
static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

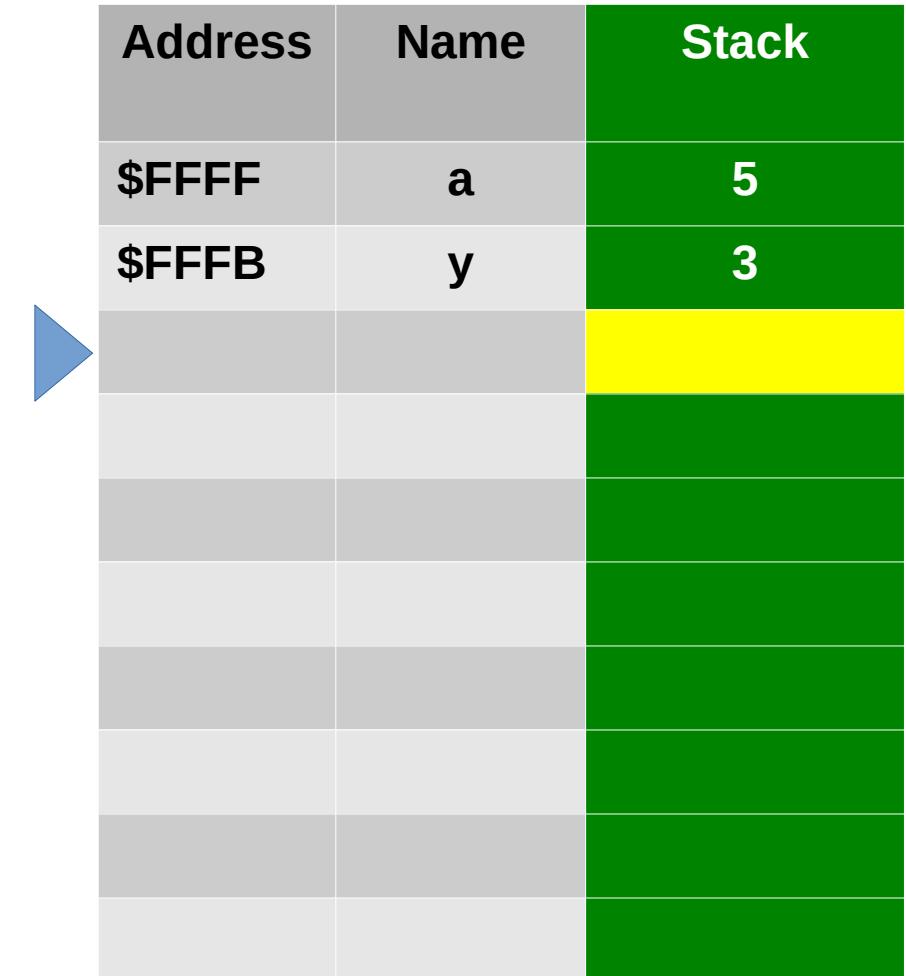
    return res;
}
```



```
static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

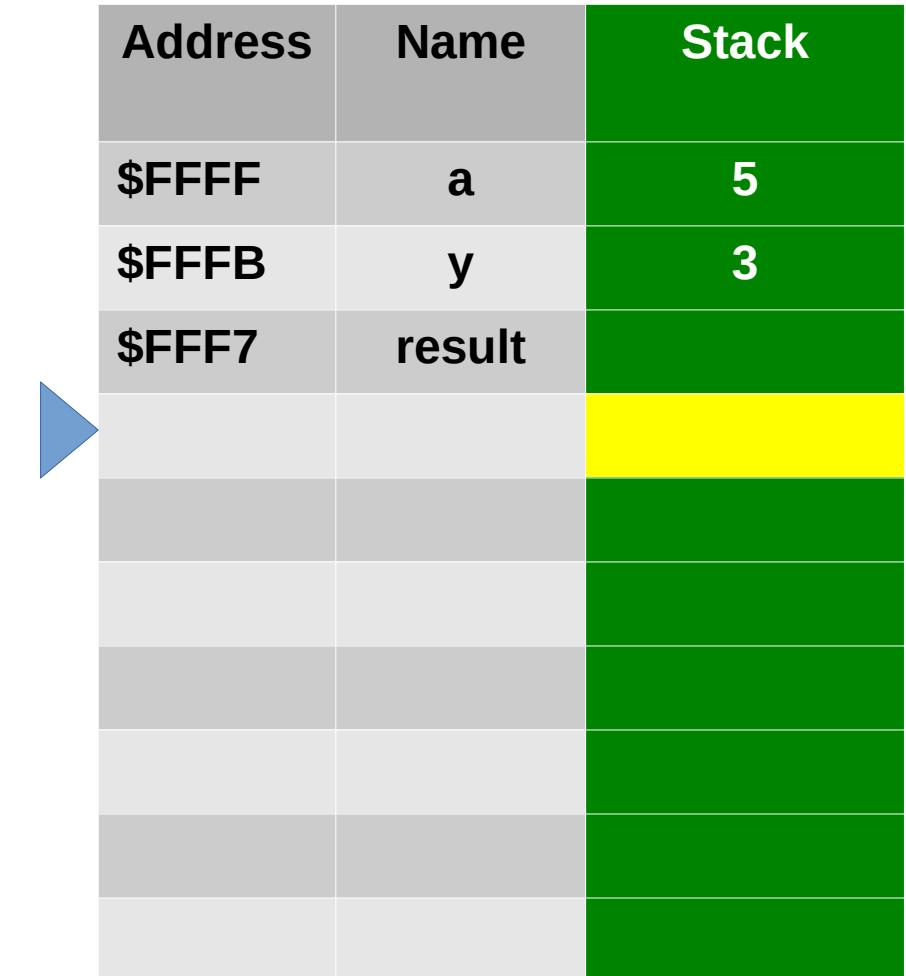
    return res;
}
```



```
static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

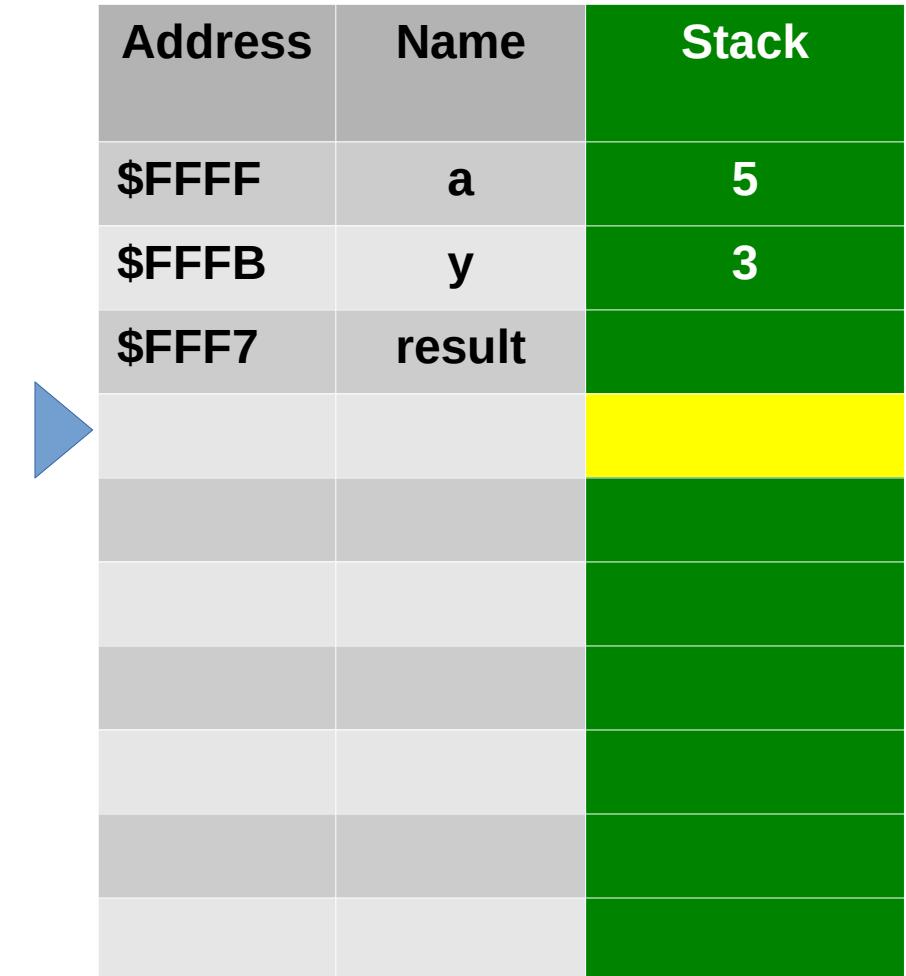
    return res;
}
```



```
static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

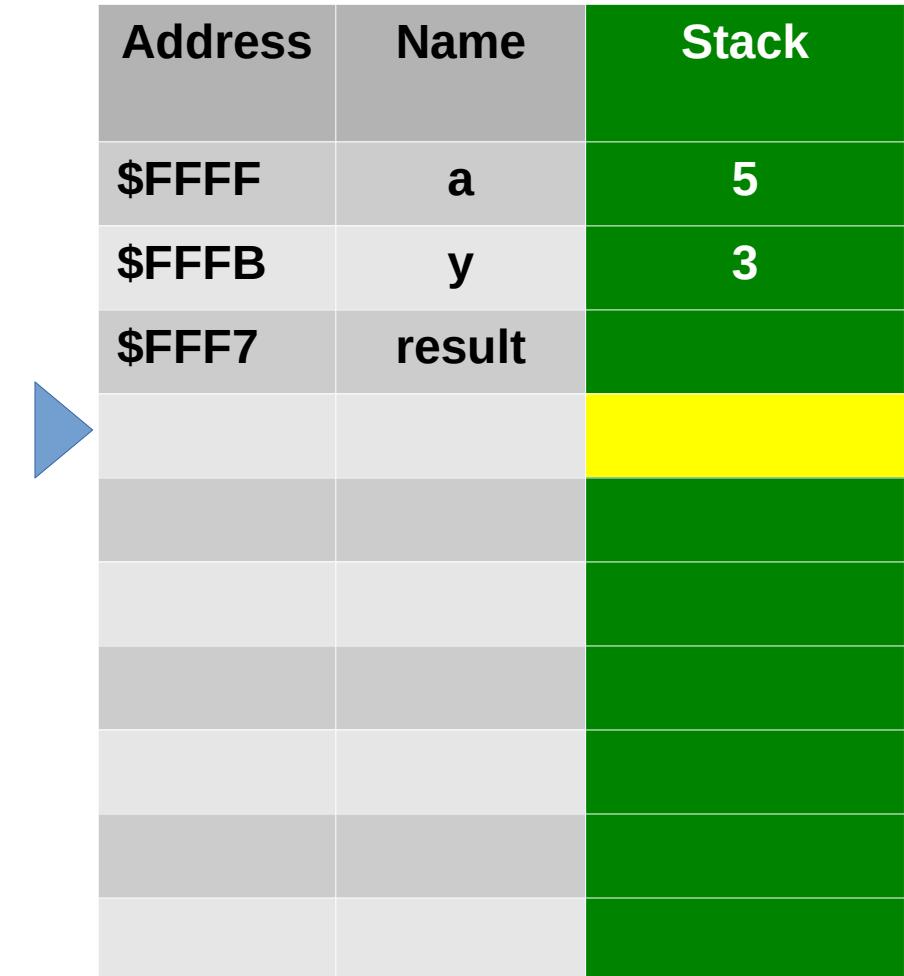
    return res;
}
```



```
static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}
```



```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```

Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	



```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```

Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	



```
static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}
```

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```

Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	



```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```

A diagram illustrating the state of the stack during the execution of the `pow` function. A blue arrow points from the variable `b` in the `pow` function's parameter list to the `result` row in the stack table, indicating that the value of `b` is being stored at that memory location.

Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```

Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3



```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```

Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	1



```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	1
\$FFE3	i	0

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	1
\$FFE3	i	0

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	1
\$FFE3	i	0

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	1
\$FFE3	i	0

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	5
\$FFE3	i	0

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	5
\$FFE3	i	0

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	5
\$FFE3	i	1

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	5
\$FFE3	i	1

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	5
\$FFE3	i	1

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```

Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	5
\$FFE3	i	1



```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	25
\$FFE3	i	1

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	25
\$FFE3	i	1

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	25
\$FFE3	i	2

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	25
\$FFE3	i	2

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	25
\$FFE3	i	2

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	25
\$FFE3	i	2

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	2

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	2

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	3

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	3

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	3

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	3

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	3

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```

Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	3

The diagram illustrates the state of the stack memory. The stack grows from high addresses to low addresses. The current stack frame starts at address \$FFF7, which contains the variable 'result' with value 125. Below it is the 'Return address' at \$FFEB, which points back to the caller's code. The local variables 'res' (value 125) and 'i' (value 3) are also present. Above the stack frame, the global variables 'a' (value 5) and 'y' (value 3) are shown. A blue arrow points to the 'Return address' cell.

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```

Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	
\$FFFA	a	5
\$FFEF	b	3
\$FFEB	Return address	
\$FFE7	res	125
\$FFE3	i	3

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```

Diagram illustrating the state of the stack during the execution of the `Main()` function.

The stack grows from high addresses (e.g., \$FFFF) to low addresses (e.g., \$FFE3). A blue arrow points upwards, indicating the direction of stack growth. An upward-pointing black arrow highlights the current stack frame for the `result` variable.

Stack State:

Address	Name	Value
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	125
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	3

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```

Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	125
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	3



```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	125
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	3

```

static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}

```



Address	Name	Stack
\$FFFF	a	5
\$FFFB	y	3
\$FFF7	result	125
\$FFF3	a	5
\$FFEF	b	3
\$FFEB	Return address	\$FFF7
\$FFE7	res	125
\$FFE3	i	3

```
static int Main()
{
    int a = 5, y = 3, result;
    result = pow(a, y);
}

static int pow(int a, int b)
{
    int res = 1;
    for(int i=0; i<b; i++)
    {
        res *= a;
    }

    return res;
}
```

