
CTEK

Пример использования стека при вызове функций

```
int factorial(int x)
{
    int a=0;
    if(x<=1)
        return x;
    a=x-1;
    return x*factorial(a);
}
void main(void)
{
    int f=0;
    f=factorial(3);
    print(f);
    ...
}
```

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←-[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←-[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	←
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code> int a=0;</code>	100: PUSH 0	
<code> if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
<code> return x;</code>	108: POP	
	110: RET	
<code> a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code> return</code>	116: PUSH r1	
<code> x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code> int f=0;</code>	200: PUSH 0	←
<code> f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code> print(f);</code>	210: PUSH r1	
	212: CALL print	
<code> ...</code>	214: POP	

Регистры

R1	0
IP	200
SP	672

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
	664	
	666	
	668	
	670	
⇒	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	←
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	0
IP	202
SP	670

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
	664	
	666	
	668	
⇒	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	←
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	0
IP	204
SP	668

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
	664	
	666	
⇒	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	←
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	0
IP	204
SP	666

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
	664	
⇒	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	←
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	←
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	0
IP	100
SP	668

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
	664	
⇒	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	←
<code>if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp-3]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	0
IP	100
SP	666

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
	664	
⇒	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	←
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	0
IP	102
SP	664

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
⇒	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	←
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	0
IP	102
SP	664

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

Кадр функции factorial()

Кадр функции main()

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	←
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	3
IP	104
SP	664

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	←
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	3
IP	106
SP	664

Flag	z=0 c=0
-------------	---------

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	←
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	3
IP	112
SP	664

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	←
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	2
IP	114
SP	664

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=0
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	←
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	2
IP	116
SP	664

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
	662	
→	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	←
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	2
IP	118
SP	662

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
	660	
⇨	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	←
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	2
IP	100
SP	660

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
	658	
⇒	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	←
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	2
IP	102
SP	658

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
⇒	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	←
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Кадр функции factorial()

Кадр функции factorial()

Кадр функции main()

Регистры

R1	2
IP	102
SP	658

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	←
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	2
IP	104
SP	658

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	←
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	2
IP	106
SP	658

Flag	z=0 c=0
-------------	---------

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code> int a=0;</code>	100: PUSH 0	
<code> if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
<code> return x;</code>	108: POP	
	110: RET	
<code> a=x-1;</code>	112: r1←r1-1	←
	114: [sp]←r1	
<code> return</code>	116: PUSH r1	
<code> x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code> int f=0;</code>	200: PUSH 0	
<code> f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code> print(f);</code>	210: PUSH r1	
	212: CALL print	
<code> ...</code>	214: POP	

Регистры

R1	2
IP	112
SP	658

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	←
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	1
IP	114
SP	658

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=0
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	←
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	1
IP	116
SP	658

Стек

SP	Адрес	Данные
	650	
	652	
	654	
	656	
→	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	←
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	1
IP	118
SP	656

Стек

SP	Адрес	Данные
	650	
	652	
	654	
→	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	←
<code>if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	1
IP	100
SP	654

Стек

SP	Адрес	Данные
	650	
	652	
⇒	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	←
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	1
IP	102
SP	652

Стек

SP	Адрес	Данные
	650	
→	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	←
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	1
IP	104
SP	652

Стек

SP	Адрес	Данные
	650	
→	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	←
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	1
IP	106
SP	652

Flag	z=1 c=0
-------------	---------

Стек

SP	Адрес	Данные
	650	
→	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	←
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	1
IP	108
SP	652

Стек

SP	Адрес	Данные
	650	
→	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	←
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	1
IP	110
SP	654

Стек

SP	Адрес	Данные
	650	
	652	a=0
→	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	←
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	←
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	1
IP	120
SP	654

Стек

SP	Адрес	Данные
	650	
	652	a=0
⇒	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	←
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	1
IP	120
SP	656

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
→	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	←
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	1
IP	122
SP	658

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
→	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	←
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	1
R2	2
IP	124
SP	658

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
⇨	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	←
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	2
R2	2
IP	126
SP	658

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
⇒	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	←
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	2
R2	2
IP	128
SP	660

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
⇒	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	←
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	2
R2	2
IP	120
SP	662

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
→	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	←
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	2
R2	2
IP	122
SP	664

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
→	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	←
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	2
R2	3
IP	124
SP	664

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
→	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
<code>int factorial(int x){</code>		
<code>int a=0;</code>	100: PUSH 0	
<code>if(x<=1)</code>	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
<code>return x;</code>	108: POP	
	110: RET	
<code>a=x-1;</code>	112: r1←r1-1	
	114: [sp]←r1	
<code>return</code>	116: PUSH r1	
<code>x*factorial(a);</code>	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	←
<code>}</code>	128: RET	
<code>void main(void) {</code>		
<code>int f=0;</code>	200: PUSH 0	
<code>f=factorial(3);</code>	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
<code>print(f);</code>	210: PUSH r1	
	212: CALL print	
<code>...</code>	214: POP	

Регистры

R1	6
R2	3
IP	126
SP	664

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
→	664	a=2
	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	←
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	6
R2	3
IP	128
SP	666

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
⇒	666	ret: 206
	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	←
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	6
R2	3
IP	206
SP	668

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
→	668	x=3
	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	←
print(f);	210: PUSH r1	
	212: CALL print	
...	214: POP	

Регистры

R1	6
R2	3
IP	208
SP	670

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
⇒	670	f=0
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	←
	212: CALL print	
...	214: POP	

Регистры

R1	6
R2	3
IP	210
SP	670

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
	668	x=3
⇒	670	f=6
	672	

Сегмент кода

Программа	Код	IP
int factorial(int x){		
int a=0;	100: PUSH 0	
if(x<=1)	102: r1←[sp+4]	
	104: CMP r1,1	
	106: JNE 112	
return x;	108: POP	
	110: RET	
a=x-1;	112: r1←r1-1	
	114: [sp]←r1	
return	116: PUSH r1	
x*factorial(a);	118: CALL 100	
	120: POP	
	122: r2←[sp+4]	
	124: r1←r1*r2	
	126: POP	
}	128: RET	
void main(void) {		
int f=0;	200: PUSH 0	
f=factorial(3);	202: PUSH 3	
	204: CALL 100	
	206: POP	
	208: [sp]←r1	
print(f);	210: PUSH r1	
	212: CALL print	←
...	214: POP	

Регистры

R1	6
R2	3
IP	212
SP	668

Стек

SP	Адрес	Данные
	650	
	652	a=0
	654	ret:120
	656	x=1
	658	a=1
	660	ret: 120
	662	x=2
	664	a=2
	666	ret: 206
⇒	668	val=6
	670	f=6
	672	