

课堂练习1--控制栈和活动记录

有如下C语言程序：

```
int i;  
int count;  
int fun (int i) {  
    count++;  
    if (i==1) return 1;  
    else return  
i*fun(i-1);  
}  
int main() {  
    i=5;  
    count=0;  
    printf( "%d",  
fun(i) );  
}
```

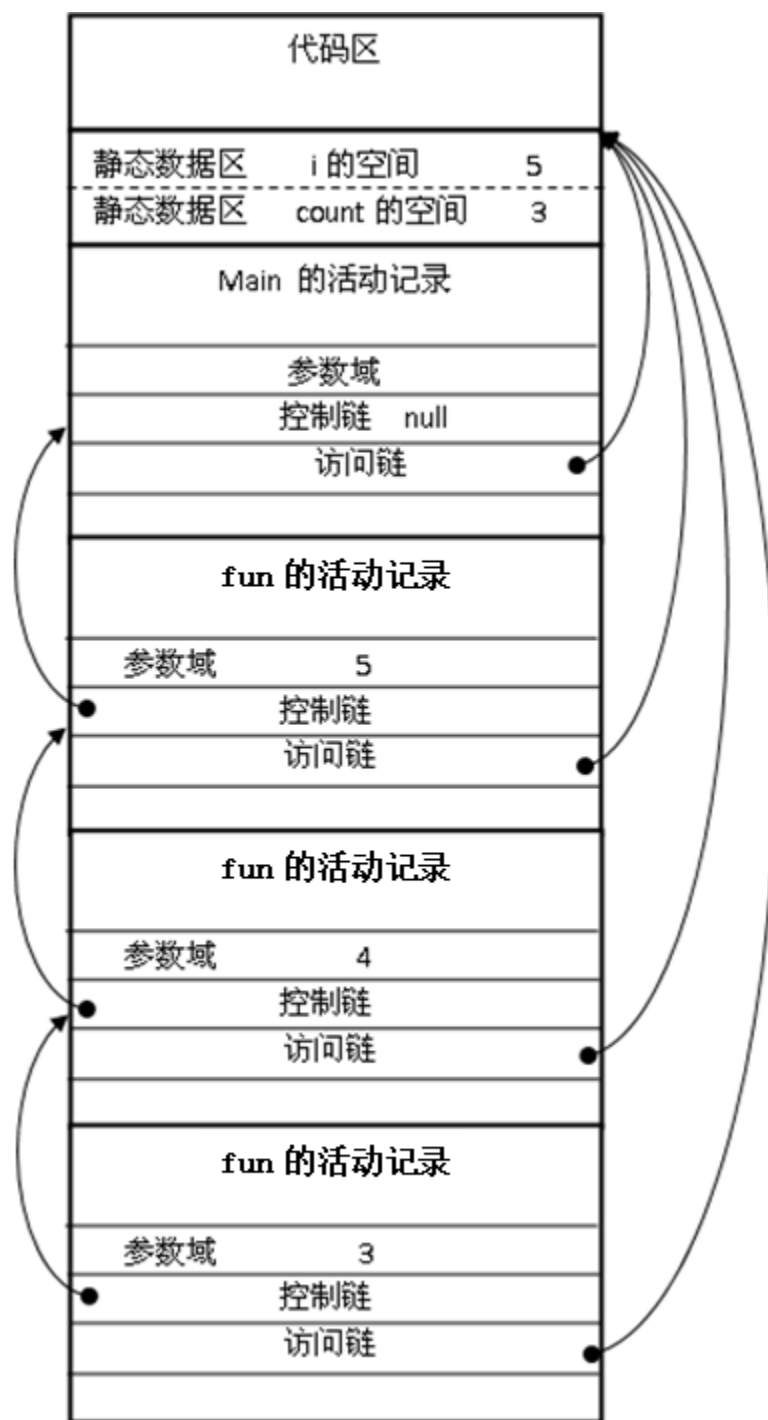
问题：

画出该程序运行时存储空间的组织示意图，并画出该程序运行过程中，当count=3时，控制栈中活动记录的示意图。

要求：

画出每个活动记录中的控制链和访问链，并给出参数域的值

解答：



练习:

```
int i;
int B[2];
void P(int m)
{
    i=0; m=m+10; B[i]=10;
    i=1; m=m+10; B[i]=10;
}
main()
{
    B[0]=10; B[1]=20;
    i=0; P(B[i]);
    printf("B[0]=d%, B[1]=d%", B[0], B[1]);
}
```

■ 传值调用

■ 引用调用

■ 复制恢复

■ 传名

解答：传值调用

```
int i;  
int B[2];  
void P(int m)  
{  
    i=0; m=m+10; B[i]=10;  
    i=1; m=m+10; B[i]=10;  
}  
main()  
{  
    B[0]=10; B[1]=20;  
    i=0; P(B[i]);  
    printf("B[0]=d%, B[1]=d%", B[0], B[1]);  
}
```

执行结果：

B[0]=10, B[1]=10

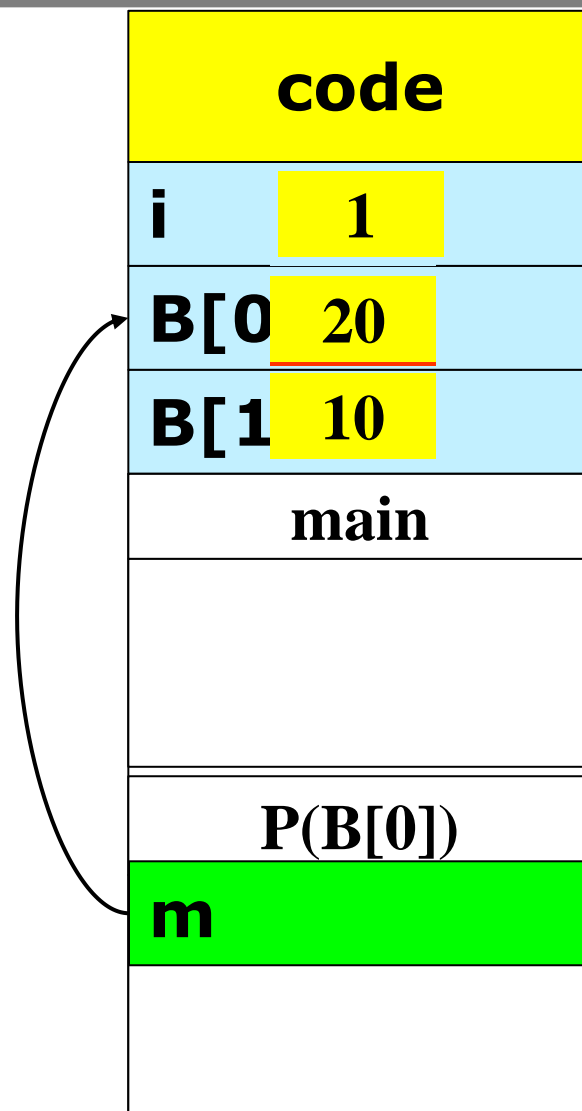
code		
i	1	
B[0]	10	
B[1]	10	
main		
P(B[0])		
m	30	

解答： 引用调用

```
int i;
int B[2];
void P(int m)
{
    i=0; m=m+10; B[i]=10;
    i=1; m=m+10; B[i]=10;
}
main()
{
    B[0]=10; B[1]=20;
    i=0; P(B[i]);
    printf("B[0]=d%, B[1]=d%", B[0], B[1]);
}
```

执行结果：

B[0]=20, B[1]=10

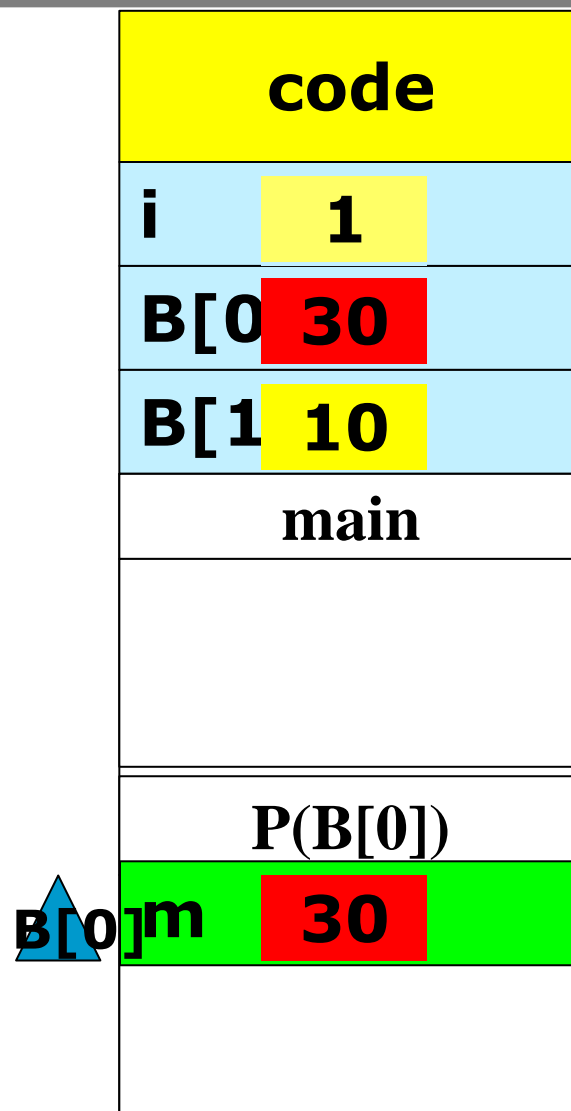


解答：复制恢复

```
int i;
int B[2];
void P(int m)
{
    i=0; m=m+10; B[i]=10;
    i=1; m=m+10; B[i]=10;
}
main()
{
    B[0]=10; B[1]=20;
    i=0; P(B[i]);
    printf("B[0]=d%, B[1]=d%", B[0], B[1]);
}
```

执行结果：

B[0]=30, B[1]=10



解答：传名调用

```
int i;  
int B[2];  
void P(int m)  
{  
    i=0; m=m+10; B[i]=10;  
    i=1; m=m+10; B[i]=10;  
}  
main()  
{  
    B[0]=10; B[1]=20;  
    i=0; P(B[i]);  
    printf("B[0]=d%, B[1]=d%", B[0], B[1]);  
}
```

执行结果：

B[0]=10, B[1]=10

**i=0;
B[i]=B[i]+1
0;
B[i]=10;
i=1;
B[i]=B[i]+1
0;
B[i]=10;**

课堂练习2--参数传递机制

有如下C语言程序：

```
int i;  
int b[4];  
void Q(int x; int y) {  
    i=1;  
    x=x+2;  
    b[i]=15;  
    y=y+3;  
    b[i]=20;  
}  
main() {  
    for (i=0;i<4;i++) b[i]=i;  
    i=1;  
    Q(b[i], b[i+1]);  
    for (i=0; i<4; i++) printf("b[%d]=%d ", &i, &b[i]);  
}
```

假定采用下面的参数传递机制，该程序的执行结果分别是什么？

- (1) 传值调用
- (2) 引用调用
- (3) 复制恢复

要求：描述程序执行过程的主要步骤。

解答：（1）传值调用

过程：

	全局变量					形参	
	i	b[0]	b[1]	b[2]	b[3]	x	y
	1	0	1	2	3		
Call Q(b[i], b[i+1])						1	2
i=1	1						
x=x+2						3	
b[i]=15			15				
y=y+3							5
b[i]=20			20				
返回							
结果	1	0		20	2	3	

结果为： b[0]=0 b[1]=20 b[2]=2 b[3]=3

解答：（2）引用调用

过程：

	全局变量					形参	
	i	b[0]	b[1]	b[2]	b[3]	x	y
	1	0	1	2	3		
Call Q(b[i], b[i+1])						b[1]	b[2]
i=1	1						
x=x+2			3				
b[i]=15			15				
y=y+3				5			
b[i]=20			20				
返回							
结果	1	0		20	5	3	

结果为： b[0]=0 b[1]=20 b[2]=5 b[3]=3

解答：（3）复制恢复

过程：

	全局变量					形参	
	i	b[0]	b[1]	b[2]	b[3]	x	y
	1	0	1	2	3		
Call Q(b[i], b[i+1])						1 / b[1]	2 / b[2]
i=1	1						
x=x+2						3	
b[i]=15			15				
y=y+3							5
b[i]=20			20				
返回			3	5			
结果	1	0		3	5	3	

结果为： b[0]=0 b[1]=3 b[2]=5 b[3]=3

课堂练习3 — 控制栈

有如下C语言程序

```
int m, n;
int count;
int func (int i) {
    int sum;
    count++;
    if (i==1) sum=1;
    else sum=i+func(i-1);
    return sum;
}
int main(void) {
    m=10;
    count=0;
    n=func (m) ;
    printf( "result=%d\n", n)
}
```

- 画出该程序运行过程中，以下时刻的逻辑地址空间组织示意图，要求给出各类数据空间的位置及状态，包括全局变量、局部变量、参数、以及返回值的空间位置及状态。

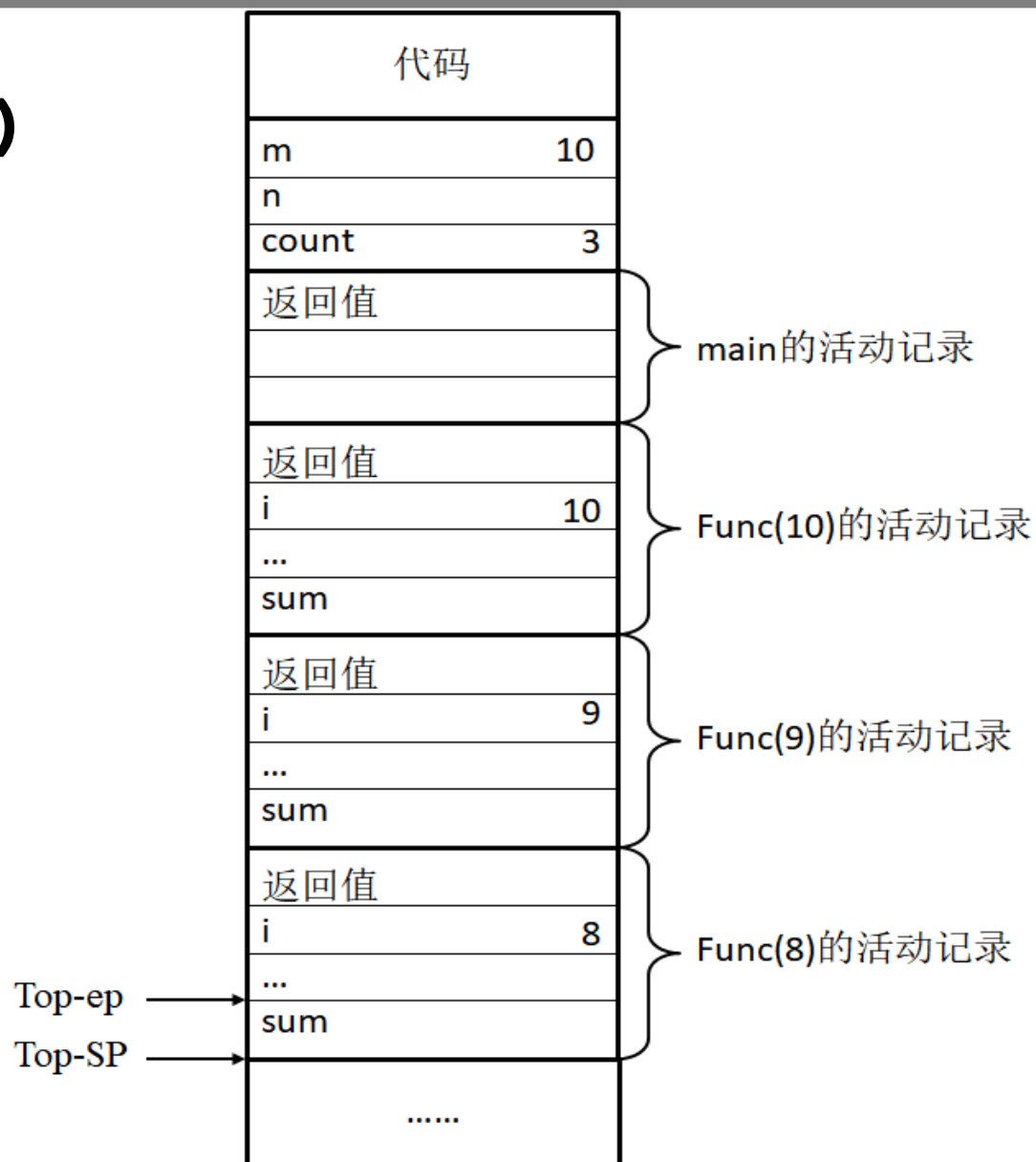
(1) 控制进入func中执行，count=3时

(2) 控制刚刚返回到函数func的第一次活动中时

(3) main函数执行语句printf(“result=d%”, n)之前

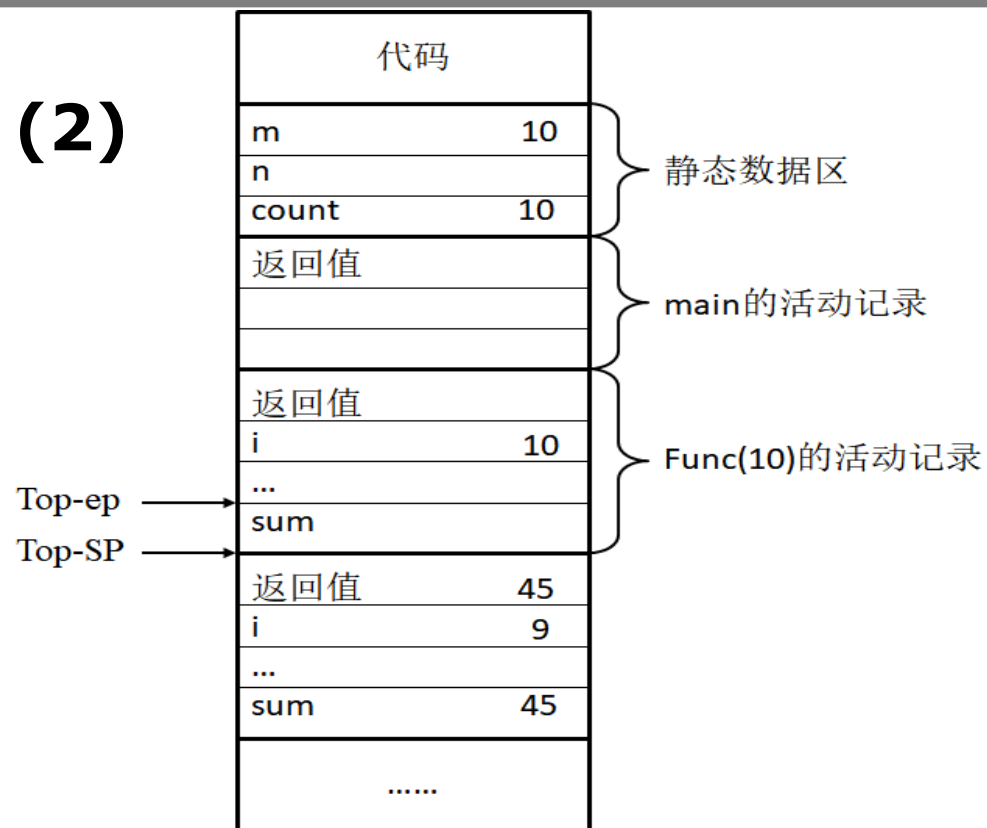
参考答案

(1)



参考答案

(2)



(3)

