Homework 7

(1）针对以下C程序片段，直接在源程序上进行循环优化（循环不变计算外提，强度消弱与复写传播优化等）

```
int a[100][100],b[100][100],c[100][100];
int i,j,k; //int : 4 bytes
for(i=0;i<100;i++)
 for(j=0;j<100;j++)
   for(k=0;k<100;k++)
    c[i][j] = c[i][j] + a[i][k] * b[k][j];
```

循环不变计算外提：

在第三层循环中，c[i][j]、a[i]不变，c[i][j]=c+i*100*4+j*4+=c+i*400+j*4，a[i]=a+i*400

```
int a[100][100],b[100][100],c[100][100];
int i,j,k; //int : 4 bytes
for(i=0;i<100;i++)
    t1 = c + i*400;
    t2 = a + i*400;
 for(j=0;j<100;j++)
    t3 = t1 + j*4;
   for(k=0;k<100;k++)
    *t3 = *t3 + t2[k] * b[k][j];
```

强度消弱：

a[i][k]=c + i*400 + 4*k 、b[k][j]= b + k*400 + j*4

```
int a[100][100],b[100][100],c[100][100];
int i,j,k; //int : 4 bytes
t1 = c;
t2 = a;
t8 = b;
for(i=0;i<100;i++)
{
    t3 = t1;    // c + i*400的初值
    t4 = t2;    // a + i*400的初值
    t5 = t3;
 for(j=0;j<100;j++)
 {
    t6 = t5;    //c + i*400 + j*4 的初值
    t7 = t4;    // a + i*400+k*4的初值
    t9 = t8;    //b + k*400 + j*4的初值
    for(k=0;k<100;k++)
   {
       *t6 = *t6 + *t7 + *t9;
        t7 = t7 + 4; // k*4
        t9 = t9 + 400; // k*400
   }
     t5 = t5 + 4; //j*4
     t8 = t8 + 4; //j*4
 }
 t1 = t1 + 400;//i*400
 t2 = t2 + 400;//i*400
}
```

复写传播优化：复写传播可以删去t3、t4

```
int a[100][100],b[100][100],c[100][100];
int i,j,k; //int : 4 bytes
t1 = c;
t2 = a;
t8 = b;
for(i=0;i<100;i++)
{
    t5 = t1;     // c + i*400的初值
 for(j=0;j<100;j++)
 {
    t6 = t5;     //c + i*400 + j*4 的初值
    t7 = t2;     // a + i*400+k*4的初值
    t9 = t8;     //b + k*400 + j*4的初值
    for(k=0;k<100;k++)
   {
       *t6 = *t6 + *t7 + *t9;
        t7 = t7 + 4; // k*4
        t9 = t9 + 400; // k*400
   }
     t5 = t5 + 4; //j*4
     t8 = t8 + 4; //j*4
 }
 t1 = t1 + 400;//i*400
 t2 = t2 + 400;//i*400
}
```

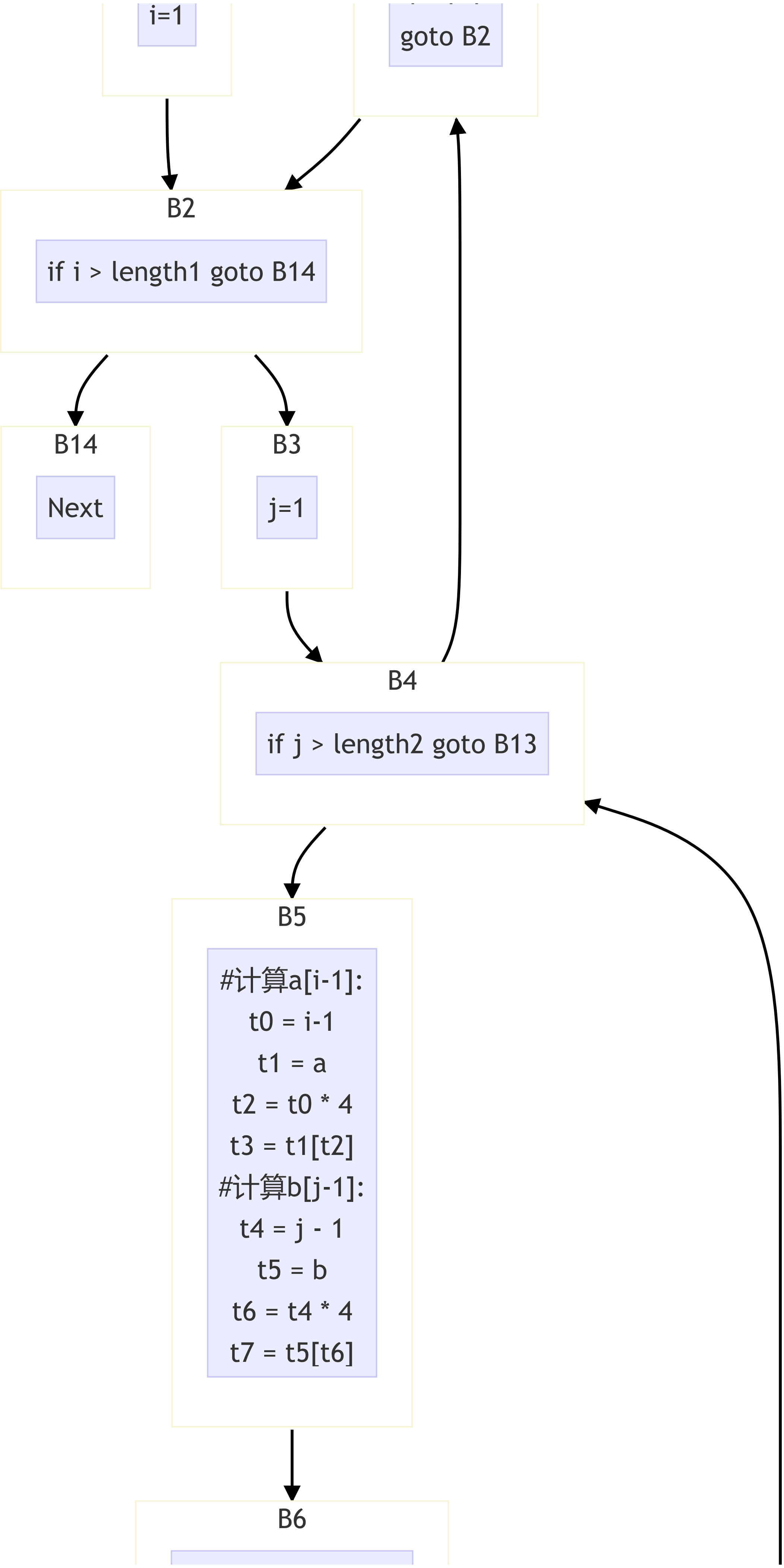
（2）针对Homework 6的（1）中的C函数，在其三地址码基础上，给出流图，回边和自然循环。

三地址码：

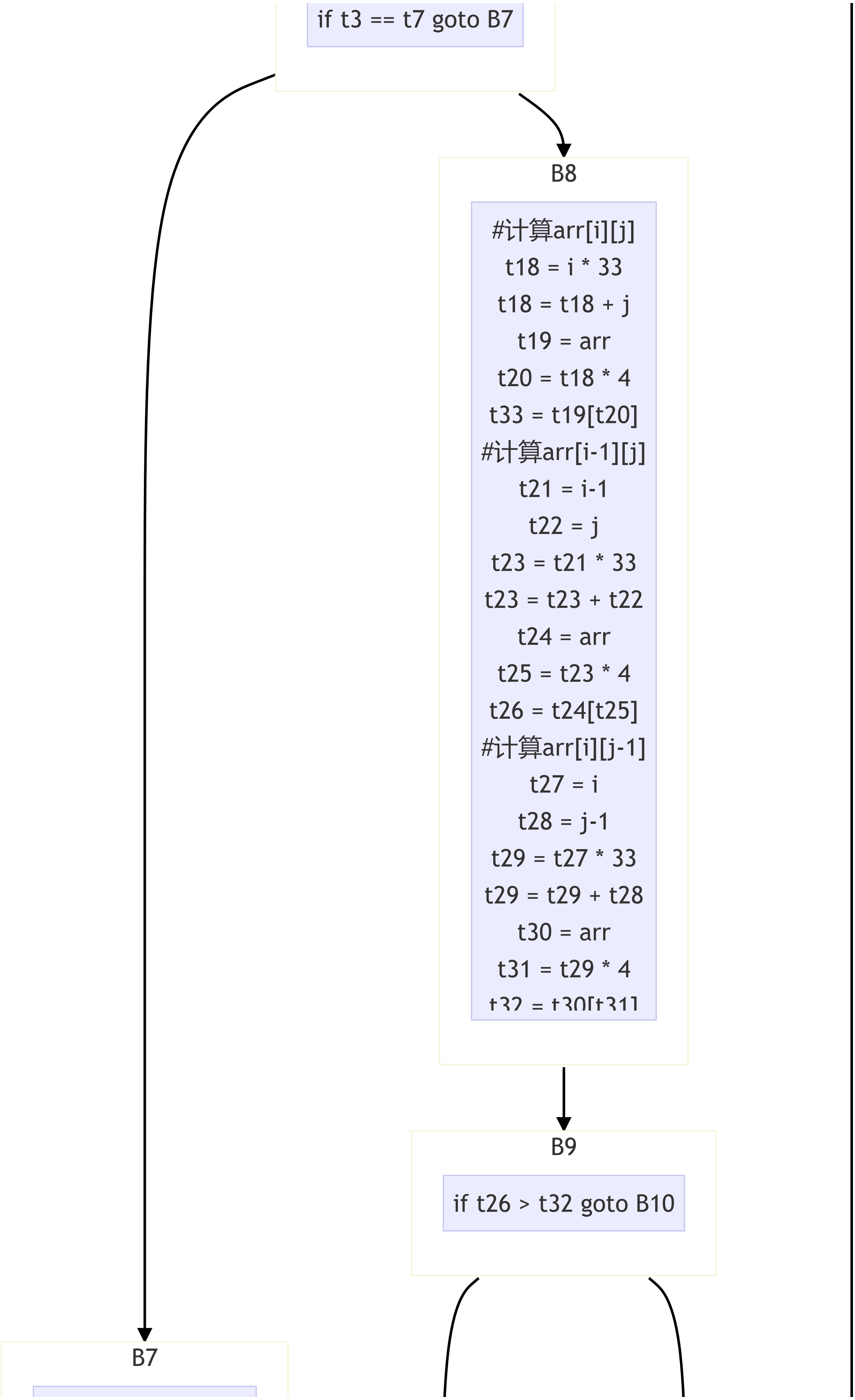
```
i=1
F1:
if i>length1 goto Fnext
j = 1
F2:
if j>length2 goto F4
#计算a[i-1]:
t0 = i-1
t1 = a
t2 = t0 * 4
t3 = t1[t2]
#计算b[j-1]:
t4 = j - 1
t5 = b
t6 = t4 * 4
t7 = t5[t6]
if t3 == t7 goto M1
goto M2
M1:
#计算arr[i,j]:
t8 = i * 33
```

```
t8 = t8 + j
t9 = arr
t10 = t8 * 4
#计算arr[i-1,j-1]+1
t11 = i-1
t12 = j-1
t13 = t11 * 33
t13 = t13 + t12
t14 = arr
t15 = t13 * 4
t16 = t14[t15]
t17 = 16 + 1
#arr[i][j] = arr[i-1,j-1]+1
t9[t10] = t17
goto F3
M2:
#计算arr[i][j]
t18 = i * 33
t18 = t18 + j
t19 = arr
t20 = t18 * 4
t33 = t19[t20]
#计算arr[i-1][j]
t21 = i-1
t22 = j
t23 = t21 * 33
t23 = t23 + t22
t24 = arr
t25 = t23 * 4
t26 = t24[t25]
#计算arr[i][j-1]
t27 = i
t28 = j-1
t29 = t27 * 33
t29 = t29 + t28
t30 = arr
t31 = t29 * 4
t32 = t30[t31]
if t26 > t32 goto L1
t33 = t32
goto F3
L1:
t33 = t26
F3:
j = j + 1
goto F2
F4:
i = i + 1
goto F1
```

流图：



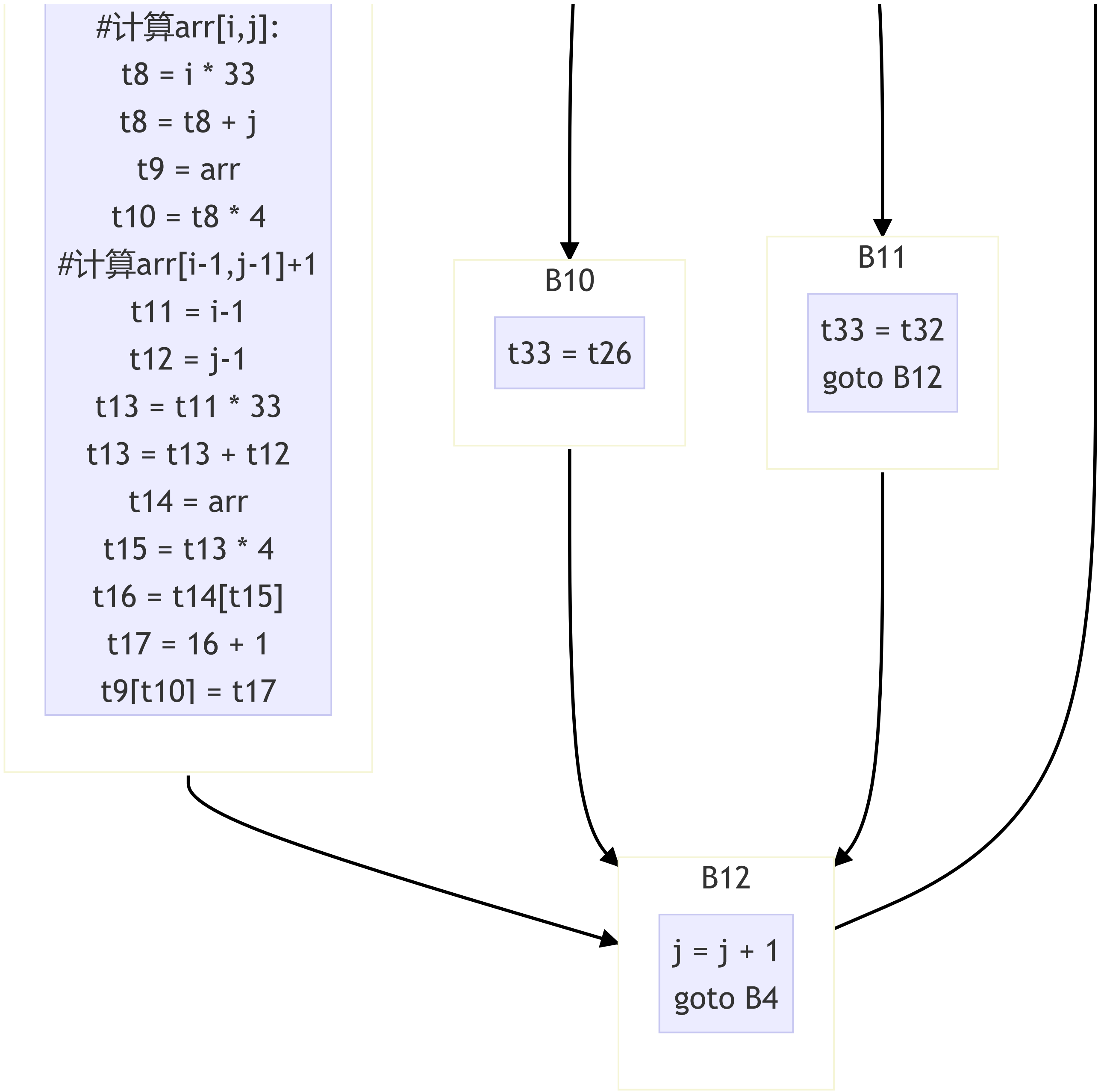
```
#计算a[i-1]:
t0 = i-1
t1 = a
t2 = t0 * 4
t3 = t1[t2]
#计算b[j-1]:
t4 = j - 1
t5 = b
t6 = t4 * 4
t7 = t5[t6]
```



```
if t3 == t7 goto B7
```

```
#计算arr[i][j]
t18 = i * 33
t18 = t18 + j
t19 = arr
t20 = t18 * 4
t33 = t19[t20]
#计算arr[i-1][j]
t21 = i-1
t22 = j
t23 = t21 * 33
t23 = t23 + t22
t24 = arr
t25 = t23 * 4
t26 = t24[t25]
#计算arr[i][j-1]
t27 = i
t28 = j-1
t29 = t27 * 33
t29 = t29 + t28
t30 = arr
t31 = t29 * 4
t32 = t30[t31]
```

```
if t26 > t32 goto B10
```



```
#计算arr[i,j]:
t8 = i * 33
t8 = t8 + j
t9 = arr
t10 = t8 * 4
#计算arr[i-1,j-1]+1
t11 = i-1
t12 = j-1
t13 = t11 * 33
t13 = t13 + t12
t14 = arr
t15 = t13 * 4
t16 = t14[t15]
t17 = 16 + 1
t9[t10] = t17
```

```
t33 = t26
```

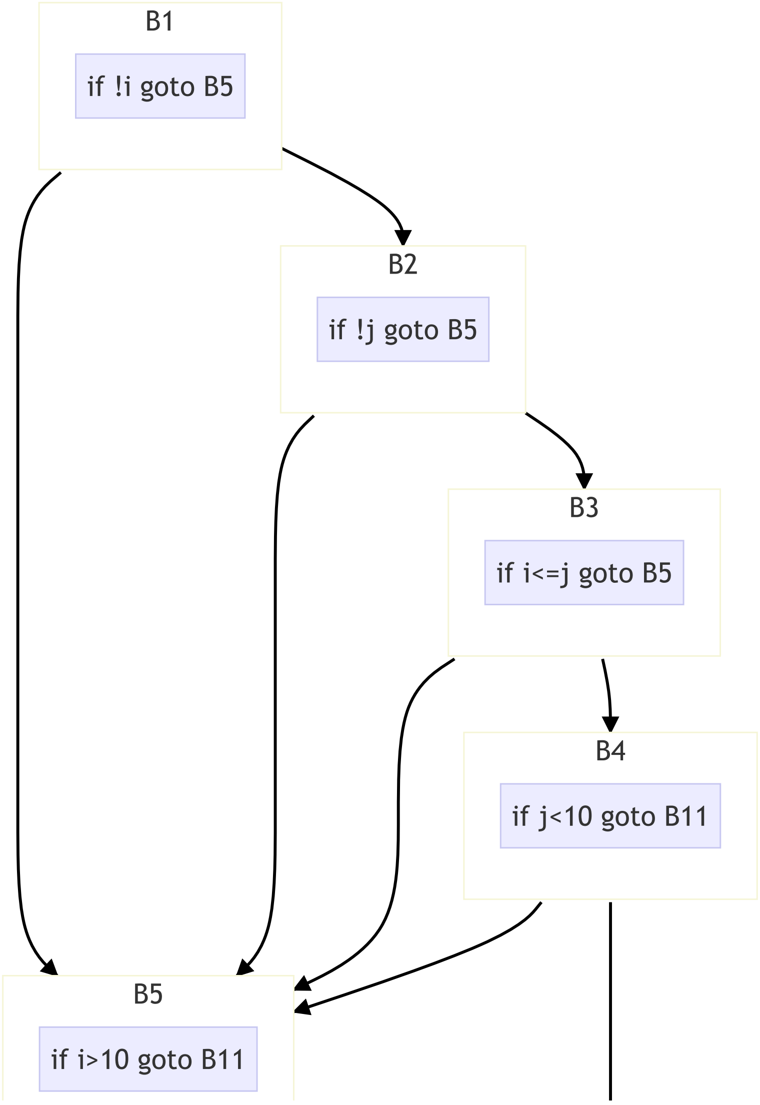
```
t33 = t32
goto B12
```

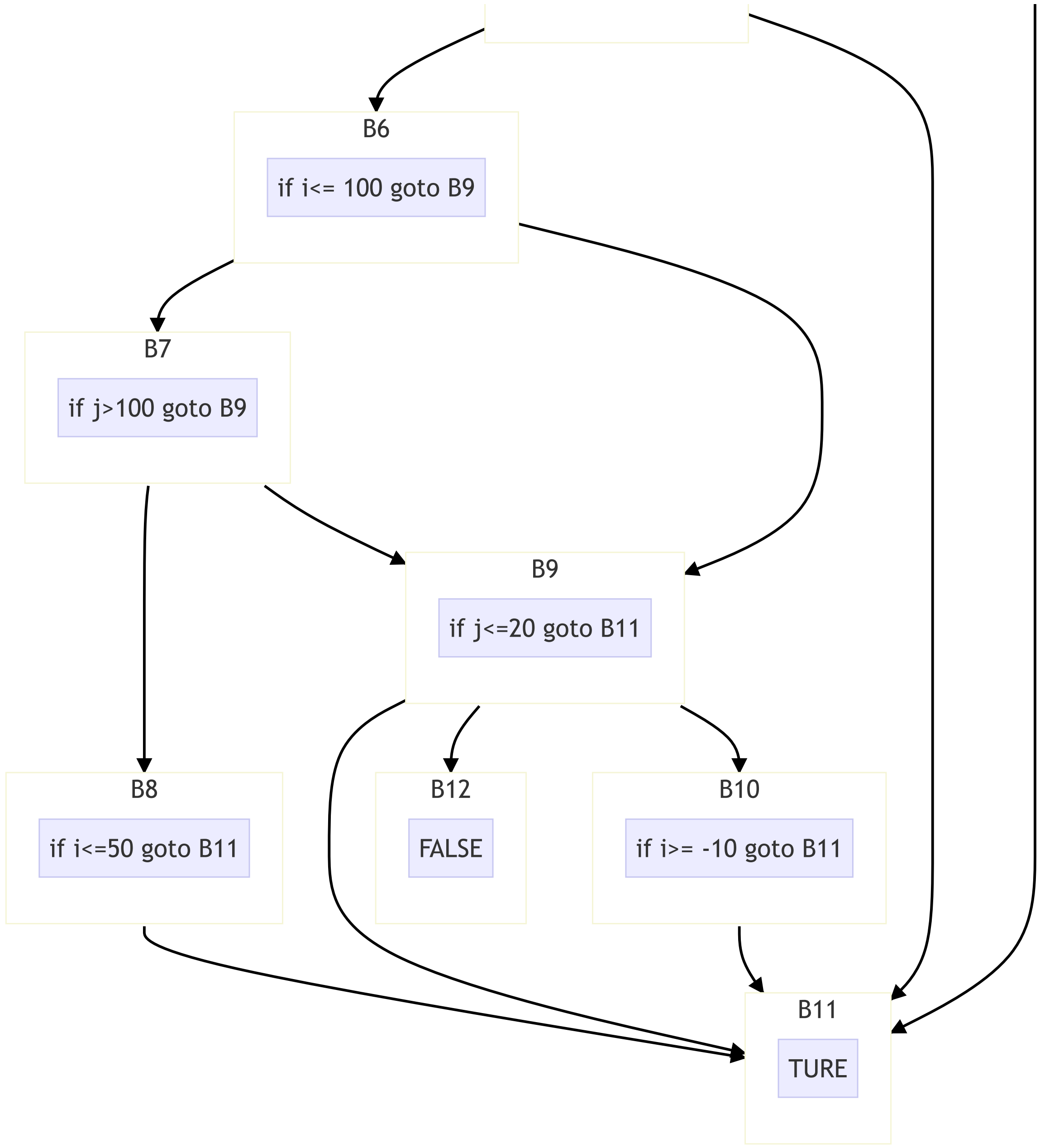
```
j = j + 1
goto B4
```

回边	自然循环
$B13 \rightarrow B2$	$\{B2, B3, B4, B5, B6, B7, B8, B9, B10, B11, B12, B13\}$
$B12 \rightarrow B4$	$\{B4, B5, B6, B7, B8, B9, B10, B11, B12\}$

（3）针对Homework 6的（2.2）中（b），在其三地址码基础上，给出基本块和流图。

```
if !i goto L0
if !j goto L0
if i<=j goto L0
if j<10 gotoT
L0:
if i>10 goto T
if i<=100 goto L1
if j>100 goto L1
if i<=50 goto T
L1:
if j<=20 goto T
if i>=-10 goto T
goto F
```





B6
if i<= 100 goto B9
B7
if j>100 goto B9
B9
if j<=20 goto B11
B8
if i<=50 goto B11
B12
FALSE
B10
if i>= -10 goto B11
B11
TURE