

T1.

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];
```

循环不变代码外提： $c[i][j]$ 在 k 循环中不变，可将其移出，在 j 循环中初始化
强度消弱： 昂贵运算替换为便宜运算
复写传播： 减少不必要的变量赋值。

优化后代码：

```
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++) {
        int temp = c[i][j]; // 循环不变代码外提
        for (k = 0; k < 100; k++) {
            temp += a[i][k] * b[k][j];
        }
        c[i][j] = temp;
    }
}
```

T2.

针对Homework 7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。

```
#define N 32
```

```
int a[N],b[N];
int arr[N+1][N+1];
```

```
void lcs()
{
```

```

for (i = 1; i <= len1; ++i){
  for (j = 1; j <= len2; ++j) {
    if (a[i - 1] == b[j - 1]) { //串中的下标从0开始
      arr[i][j] = arr[i - 1][j - 1] + 1;
    }
    else {
      arr[i][j] = arr[i - 1][j] > arr[i][j - 1] ? arr[i - 1][j] : arr[i][j - 1];
    }
  }
}
} // end of lcs()

```

三地址码:

```

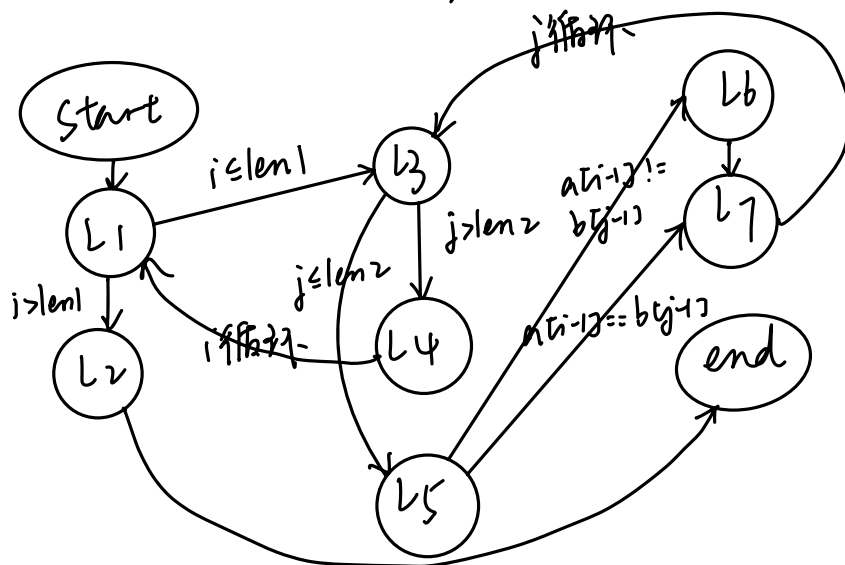
lcs: // 函数开始
  i = 1
L1: // 外层循环的开始
  if i > len1 goto L2
  j = 1
L3: // 内层循环的开始
  if j > len2 goto L4
  t1 = i - 1
  t2 = j - 1
  t3 = a[t1]
  t4 = b[t2]
  if t3 == t4 goto L5
  t5 = i - 1
  t6 = arr[t5][j]
  t7 = i
  t8 = j - 1
  t9 = arr[t7][t8]
  if t6 > t9 goto L6
  arr[i][j] = t9
  goto L7
L5: // 当 a[i-1] == b[j-1]
  t10 = i - 1
  t11 = j - 1
  t12 = arr[t10][t11]
  t13 = t12 + 1
  arr[i][j] = t13
  goto L7
L6: // 当 arr[i-1][j] > arr[i][j-1]
  arr[i][j] = t6
L7: // 结束 if-else 块
  j = j + 1
  goto L3
L4: // 结束内层循环
  i = i + 1
  goto L1
L2: // 结束外层循环
  return

```

基本块: 每个标签和函数的开始和结束视为基本块边界。

1. 边: goto L1
goto L3

自然循环: 由边形成的循环。



T3.

针对Homework 7的 (2.2) 中 (b) , 在其三地址码基础上, 给出基本块和流图。

三地址代码:

- (1) ifFalse a goto (10)
- (2) if b goto (4)
- (3) ifFalse c goto (10)
- (4) if d goto (6)
- (5) ifFalse e goto (7)
- (6) ifFalse f goto (10)
- (7) if g goto (9)
- (8) ifFalse h goto (10)
- (9) if i goto (11)
- (10) ifFalse j goto (12)
- (11) // 表达式为真
- (12) // 表达式为假

