

# 23级卓中卓选拔初试机考任务

## 一、12位整数编码

- 用12位的存储空间表示整数，一般只能有 $2^{12}=4096$ 个数字。现采用如下编码方式：
- 用8位表示一个常数 `immed_8`，用4位表示该常数要进行移位操作的移位数 `rotate_imm`，最终这12位表示的数值是：`immed_8` 循环右移 `rotate_imm*2` 位后所得到的值。
- 示例：`0x3fc`可以由8位常数`0xff`循环右移30位得到，是常数表达式。  
示例：`0x1fe`，虽然可以由`0xff`循环右移31位得到，但是移动的位数不是偶数，因此不符合该编码方式。
- 如果只考虑正整数，请编写程序，
  - 该程序从标准输入读取一个非负32位十进制整数 `K`，
  - 如果 `K` 可以采用此编码方式，则输出 `-1`，程序结束；
  - 如果不行，该程序将试着找出两个离 `K` 最近的两个数 `m` 和 `n`，要求：
    - `m` 比 `K` 小，且 `m` 及 `K-m` 都可以采用上述编码；
    - `n` 比 `K` 大，且 `n` 及 `K-n` 都可以采用上述编码；
- 如果 `m` 和 `n` 都找到，则输出 `K`，`m` 和 `n`，格式如下：

```
1 K,m,n
```

- 即输出的三个数用英文逗号隔开；
- 否则输出 `-2`。

## 已给代码

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 int main(){
5
6 }
```

## 二、图中是否有环

- 定义一个带权无向图的数据结构如下：

```
1 typedef struct GraphStruct{
2     int vertex_num;
3     int edge_num;
4     int **edges;
5 }GraphStruct;
6 typedef struct GraphStruct *Graph;
7 typedef unsigned int Vertex;
8 typedef struct {
9     Vertex v;
10    Vertex w;
11    int weight;
12 } Edge;
```

- 其中 edges 指向图的邻接矩阵，矩阵元素为对应边上的权重(权重为自然数)，请使用该数据结构，从标准输入读入一个图，并判断该图是否有环，如果有环，则输出yes，否则输出no。
- 输入数据示例如下：

```
1 15,7
2 0,7,69
3 1,10,68
4 3,9,71
5 3,12,92
6 4,6,71
7 4,8,6
8 5,10,83
```

- 其中第 1 行分别为图中结点的个数 15、图中边的个数 7；之后的7行分别是各边的信息，以第2行为例，其表示结点0和7之间有权重为69的无向边。

## 已给代码

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 typedef struct GraphStruct{
5     int vertex_num;
```

```

6     int    edge_num;
7     int    **edges;
8 }GraphStruct;
9 typedef struct GraphStruct *Graph;
10 typedef unsigned int Vertex;
11
12
13 int main(){
14
15 }

```

### 三、图中的支配关系

- 定义一个带权有向图的数据结构如下：

```

1 typedef struct GraphStruct{
2     int    vertex_num;
3     int    edge_num;
4     int    **edges;
5 } GraphStruct;
6 typedef struct GraphStruct *Graph;
7 typedef unsigned int Vertex;
8 typedef struct {
9     Vertex v;
10    Vertex w;
11    int weight;
12 } Edge;

```

- 其中 edges 指向图的邻接矩阵，矩阵元素为对应边上的权重(权重为自然数)。  
定义图中0结点为入口结点；  
定义dom关系为：如果从图的入口结点（即0结点）到结点 n 的每条路径都经过结点 d ，则称结点 d 支配(dominate) n ，记为 d dom n。  
根据该定义，每个结点支配它自己，且入口结点（即0结点）支配包括它自己在内的所有结点。
- 请使用上述数据结构，从标准输入读入一个图(该图为联通图)，请找出除入口结点外所有结点间的支配关系，并按下面示例样式输出：
- 输入数据示例如下：

```

1 5,6
2 0,1,95
3 0,2,41
4 1,3,99

```

```
5  1,4,67
6  2,4,95
7  3,4,99
```

- 其中第1行分别为图中结点的个数 5、图中边的个数 6；之后的6行分别是各边的信息，以第2行为例，其表示结点0到1之间有权重为95的有向边，方向为0至1。根据支配关系的定义，该图的支配关系有：

```
1  0 dom 0
2  0 dom 1
3  0 dom 2
4  0 dom 3
5  0 dom 4
6  1 dom 1
7  1 dom 3
8  2 dom 2
9  3 dom 3
10 4 dom 4
```

- 因0结点支配所有结点，所以不输出该信息；将其它支配关系按结点编号从小到大的顺序输出如下：

```
1  (1#1)(1#3)(2#2)(3#3)(4#4)
```

- 其中每一对括号表示一个支配关系，# 表示 dom。

## 已给代码

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  typedef struct GraphStruct{
5      int vertex_num;
6      int edge_num;
7      int **edges;
8  }GraphStruct;
9  typedef struct GraphStruct *Graph;
10 typedef unsigned int Vertex;
11
12
```

```
13 int main(){  
14  
15 }
```