

## 闰年

实现满足下面功能的汇编程序：

输入一个年份  $n$ ，判断  $n$  是否为闰年。

### 输入格式

输入一个整数  $n$ 。

### 输出格式

输出 0 或者 1。输出 0 代表  $n$  不是闰年，输出 1 代表  $n$  是闰年。

### 约定

- 1、 $0 < n < 2147483647$
- 2、请勿使用 `.globl main`
- 3、请使用 `syscall` 结束程序：

```
li $v0,10
```

```
syscall
```

### 输入样例

```
1900
```

### 输出样例

```
0
```

### 输入样例

```
2004
```

### 输出样例

```
1
```

### 提交要求

- 请勿使用 `.globl main`。
- 不考虑延迟槽。
- 只需要提交 `.asm` 文件。
- 程序的初始地址设置（Mars->Settings->Memory Configuration）为 **Compact, Data at Address 0**。

## 矩阵转化

实现满足下面功能的汇编程序：

输入一个  $n$  乘  $m$  的稀疏矩阵  $A$ （矩阵每个元素为占一个字的整数），将  $A$  转化为三元组列表(该列表的排列顺序为：行号小的在前，如果行号相同则列号小的在前)，并将三元组列表逆序输出。

### 输入格式

第一行是一个整数  $n$ ,第二行是一个整数  $m$ 。接下来的  $n$  乘  $m$  行每行一个整数，矩阵  $A$  的第  $a$  行，第  $b$  列的元素，为上述输入的第  $(a-1)*m+b$  个整数（即一行一行地输入矩阵  $A$  的每一个元素）。

### 输出格式

$x$  行，按照输入顺序的逆序输出  $x$  个非 0 元素的信息：每行输出 3 个整数，依次为矩阵非 0 元素对应的行数，列数和数值，中间以空格隔开。

### 约定

- 1、 $0 < n < 50$
- 2、 $0 < m < 50$
- 3、矩阵每个元素值占一个整数
- 4、请勿使用 `.globl main`
- 5、请使用 `syscall` 结束程序：

```
li $v0,10
```

```
syscall
```

### 输入样例

```
2
4
1
0
0
```

```
0
5
0
0
0
```

输出样例

```
2 1 5
1 1 1
```

提交要求

- 请勿使用 `.globl main`。
- 不考虑延迟槽。
- 只需要提交 `.asm` 文件。
- 程序的初始地址设置（Mars->Settings->Memory Configuration）为 **Compact, Data at Address 0**。

## 哈密顿回路

实现满足下面功能的汇编程序：

输入一个具有  $n$  个顶点的无向图  $G$ ，判断  $G$  是否有哈密顿回路。（哈密顿回路问题，建议使用递归解决）

### 输入格式

第一行是一个整数  $n$ ，代表  $G$  有  $n$  个顶点，第二行是一个整数  $m$ ，代表  $G$  有  $m$  条边，接下来的  $2*m$  行，每行具有一个整数，设每个奇数行的数为  $a$ ，它下一行的数  $b$ ，序号为  $a,b$  的两个顶点间具有一条边，两个整数之间以回车隔开（点的标号从 1 开始）

### 输出格式

输出一个整数，若为 0 则代表  $G$  不具有哈密顿回路，若为 1 则代表  $G$  具有哈密顿回路。

### 约定

- 1、 $0 < n < 8$
- 2、 $0 < m < 100$
- 3、请勿使用 `.globl main`
- 4、最大运行指令条数限制为 100000
- 5、请使用 `syscall` 结束程序：

```
li $v0,10
```

```
syscall
```

### 输入样例

```
5
6
1
2
1
```

```
3
2
3
2
4
3
5
4
5
```

输出样例

```
1
```

输入样例

```
5
6
1
2
1
3
2
3
2
4
1
4
4
5
```

输出样例

```
0
```

提交要求

- 请勿使用 `.globl main`。
- 不考虑延迟槽。
- 只需要提交 `.asm` 文件。
- 程序的初始地址设置（Mars->Settings->Memory Configuration）为 **Compact, Data at Address 0**。