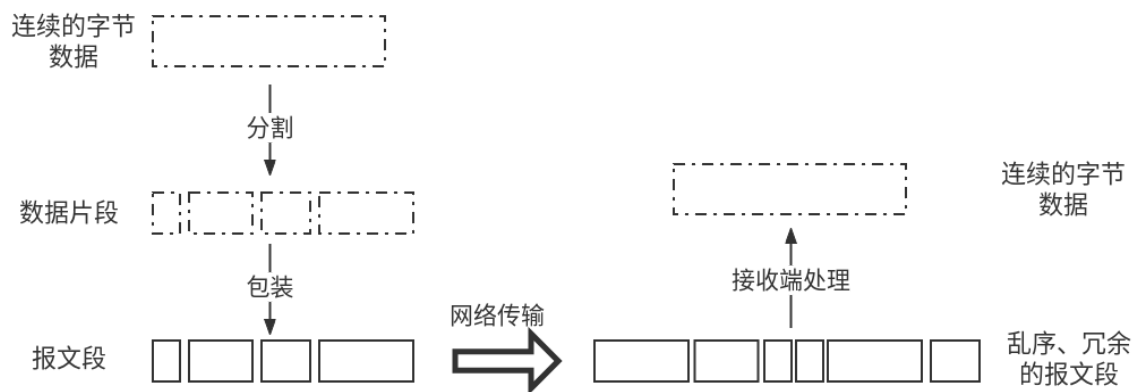


# TCP Segment

传输控制协议（TCP）是一种面向连接的、可靠的、基于字节流的传输层通信协议。在TCP传输中，一段连续的字节数据会被切成许多部分，装入到报文段（segment）中，发送给TCP接收端。我们需要实现一个简化的TCP接收端程序，接收报文段，将其还原成有序完整的原始字节数据并输出。



一个简化的TCP Segment由以下两部分组成：

部分	变量类型	描述
<b>payload</b>	字符串	是segment携带的一段数据，数据中包含一个或多个字节，其中每个字节都对应于一个序号
<b>seqno</b>	32位无符号整型数	代表 <b>payload</b> 携带数据中，第一个字节的序号

## 需求描述

**需求1.**将segment中包含的**seqno**（Sequence Number）还原成**absno**（Absolute Number）：

- 为了防止数据冲突和一些安全隐患，发送端并不会从0开始生成seqno，而是使用一个随机值**ISN**作为seqno的初始值，后续字节对应的seqno在ISN的基础上递增
- seqno是字节经由发送端处理后的序号，从ISN开始
- absno是字节在原始数据中的序号，从1开始
- 发送端在发送数据前，会将absno转换为seqno，而我们实现的接收端需要seqno还原为absno
- 用例中所有字节的seqno的范围为[0, L)

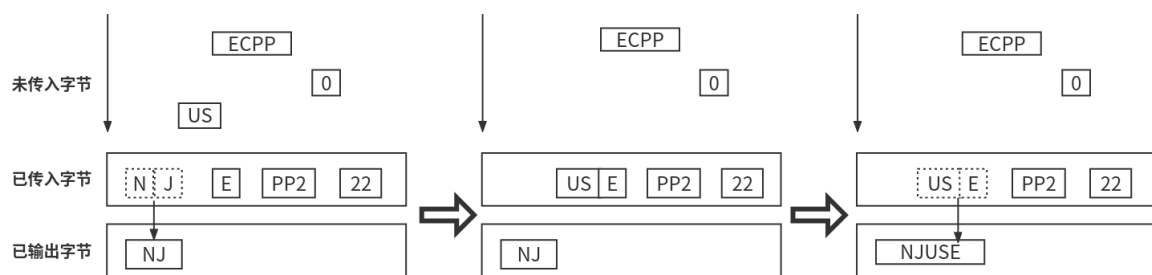
假设原始数据是“CPP22”,那么每一位字节、seqno、absno的对应关系如表所示：

字节	C	P	P	2	2
seqno	ISN	ISN+1	ISN+2	ISN+3	ISN+4
absno	1	2	3	4	5

**需求2.**实现接收端缓冲区，接收字节数据，将其还原并尽快输出，同时处理segment乱序发送、重复发送、字符数据重叠、数据越界的情况：

- 缓冲区：暂存收到的字节数据，缓冲区容量为**C**
- 尽快输出：
  - 假设经过一次输入后，已输出数据的absno范围为  $[x1, x2)$ ，缓冲区内现存数据absno范围为  $[x2, x3)$ ,  $[x3, x4)$ ,  $[x5, x6]$ ，其中  $x4 < x5$ ，则本轮输出的absno范围为  $[x2, x4)$
- 处理乱序发送：
  - 发送端发送的数据可以任意次序到达接收端
- 处理重复发送：
  - 假设本轮新接收数据的absno范围为  $[x, y)$ ，而已输出或已接收到的数据中，已经包含范围  $[x, y)$ ，则舍弃本轮新接收的数据
- 处理字符数据重叠：
  - 假设已输出数据序列为  $\langle x1, x2, x3 \rangle$ ，本轮新接收的数据序列为  $\langle x2, x3, x4 \rangle$ ，则实际接收到的字符数据序列为  $\langle x4 \rangle$
  - 假设已接收的数据序列为  $\langle x1, x2, x3 \rangle$ ，本轮新接收的数据序列为  $\langle x2, x3, x4 \rangle$ ，则将接收的字符数据序列更新为  $\langle x1, x2, x3, x4 \rangle$
- 处理数据越界：
  - 假设本轮新接受的absno数据序列为  $\langle x1, x2, x3, x4 \rangle$ ，其中  $x1 > C$ ，则丢弃这个序列
  - 假设本轮新接受的absno数据序列为  $\langle x1, x2, x3, x4 \rangle$ ，其中  $x3 = C$ ，则丢弃缓冲区外的数据，即将接收的字符数据序列更新为  $\langle x1, x2, x3 \rangle$
- 参考下图案例

假设原始数据是“NJUSECPP2022”，传入以及输出模拟示意图：



(附加) **需求3.**处理seqno的溢出

- 仅有**10%**的数据会发生seqno溢出的情况

- 由于 $0 \leq \text{seqno} < L$ ，且 $\text{seqno}$ 从 $\text{ISN}$ 开始递增，当原始数据过长时可能会导致 $\text{seqno}$ 溢出
- 当 $\text{seqno}$ 将要溢出，会重新从0开始累加

假设原始数据是“CPP22”, $\text{ISN} = L - 2$ , 那么每一位字节、 $\text{seqno}$ 、 $\text{absno}$ 的对应关系如表所示：

字节	C	P	P	2	2
<b>seqno</b>	L-2	L-1	0	1	2
<b>absno</b>	1	2	3	4	5

## 输入描述

第1行为两个正整数**N**，**C**，**ISN**，**L**。**N**为后续输入的行数，**C**为缓冲区容量大小，**ISN**为 $\text{seqno}$ 的起始值，**L-1**为 $\text{seqno}$ 的最大值

第2~N+1行每行依次输入两个变量 **seqno**，**payload**，两个变量之间以空格分隔

- $1 \leq N \leq 100$ ， $1 \leq C \leq 2000$ ， $10 \leq \text{ISN} < L \leq 2000$ ， $1 \leq \text{absno} \leq 2000$
- 保证位置重叠的字符一定相同，输入数据不含'#'字符
- 保证一个 $\text{seqno}$ 仅对应一个 $\text{absno}$

## 输出描述

从第2行输入开始，每输入一次，就需要进行一次输出。

- 每次输出三个变量**absnum**，**bytes**，**buffer**，三个变量之间以空格分隔
- 若本轮输入后满足输出条件，将可连续输出的数据全部放入字符串**bytes**中输出，**absnum**为**bytes**的第一位字节对应的 $\text{absno}$
- **buffer**是一个长度为**C**的字符串，代表本轮输入输出后的缓冲区在 $1 \leq \text{absno} \leq C$ 位置上暂存的字符情况，若在某位置上不存在暂存字符，则置为"**#**"
- 若本轮输入后不满足输出条件，则直接输出**absnum**为0，**bytes**为“null”，**buffer**正常输出

## 示例

### 示例1

输入

abc

```
3 5 4 8
6 c
4 a
5 b
```

## 输出

```
0 null ##c##
1 a ##c##
2 bc #####
```

## 解释

- 原始数据为"abc"
- 输入3行 (**N**), 缓冲区大小为5 (**C**), seqno从4开始 (**ISN**), seqno的值应小于8 (**L**)
- 6 c 对应的absno为3, 暂存, 无输出
- 4 a 对应的absno为1, 满足输出条件
- 5 b 对应的absno为2, 满足输出条件

## 示例2

### 输入

```
6 14 536 2000
536 nju
544 2022
548 2202
542 p
541 cpp
537 jusec
```

### 输出

```
1 nju #####
0 null #####2022##
0 null #####202222
0 null #####p#202222
0 null #####cpp202222
4 secpp202222 #####
```

## 解释

- 原始数据为"njusecpp202222"
- 输入6行 (N), 缓冲区大小为14 (C), seqno从536开始 (ISN), seqno的值应小于2000 (L)
- 536 nju 对应的absno为1, 满足输出条件
- 544 2022 对应的absno为9, 暂存
- 548 2202 部分超出缓冲区范围, 按照缓冲区容量进行截断, 并暂存前两个字节
- 542 p 对应的absno为7, 暂存
- 541 cpp 对应的absno为6, 暂存, 并覆盖542 p的结果
- 537 jusec 对应的absno为2, 满足输出条件

## 示例3

### 输入

```
7 11 21 100
22 cps
34 et
24 s
30 t
32 t
24 segmen
21 t
```

### 输出

```
0 null #cps#####
0 null #cps#####
0 null #cps#####
0 null #cps#####t#
0 null #cps#####t#
0 null #cpsegment#
1 tcpsegment #####
```

## 解释

- 原始数据为"tcpsegment"
- 输入7行 (N), 缓冲区大小为11 (C), seqno从21开始 (ISN), seqno的值应小于100 (L)
- 22 cps 对应的absno为2, 暂存, 无输出
- 34 et 超出缓冲区范围, 丢弃
- 24 s 对应的absno为4, 暂存, 无输出, 并覆盖22 cps的最后一个字符
- 30 t 对应的absno为10, 暂存, 无输出

- 32 t 超出缓冲区范围，丢弃
- 24 segmen 对应的absno为4，暂存，无输出
- 21 t 对应的absno为1，满足输出条件