# C++ Test

## Objective

You are provided with a packet capture file in a proprietary format with OUCH protocol packets strewn across multiple TCP streams. Your goal is to write a program to count the total number of each **OUCH Message Type** across each stream, and for the **Executed Message** sum the ExecutedShares field across each stream, and display them as output at the end of the program.

The general format of the output should match something like below

|  |
| --- |
| Stream 4  Accepted: 1330 messages  System Event: 0 messages  Replaced: 0 messages  Canceled: 4129 messages  Executed: 2 messages: 700 executed shares  …  Stream 3  Accepted: 8209 messages  System Event: 0 messages  Replaced: 0 messages  Canceled: 8432 messages  Executed: 4 messages: 346 executed shares  Totals:  Accepted: 40069 messages  System Event: 1 messages  Replaced: 1 messages  Canceled: 39917 messages  Executed: 10 messages: 3251 executed shares |

## Input File

You are provided with the input file OUCHLMM2.incoming.packets, consisting of packet capture in a proprietary format. The file itself does not have a header, but consists of records each representing a packet capture.

## Packet Capture Header

Each packet is framed using the following header:

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset (bytes)** | **Size (bytes)** | **Type** | **Description** |
| 0 | 2 | Unsigned Integer | Stream Identifier |
| 2 | 4 | Unsigned Integer | Packet Length |

All integers in the packet capture header are Big Endian.

## Packets

The packets that are captured contain within them OUCH protocol message streams. Each packet either contains the full OUCH protocol message or a partial OUCH protocol message that is completed by a subsequent packet that belongs to the same stream.

## OUCH Protocol

The OUCH protocol messages provided to you in the capture consist of following message types. Once again all the integer fields in the capture are Big Endian. The offset below are independent of the packet capture header above~~.~~

### System Event Message

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset (bytes)** | **Size (bytes)** | **Type** | **Description** |
| 0 | 2 | Unsigned Integer | OUCH Message Length (excluding this field)  [expect 11] |
| 2 | 1 | Char | OUCH Packet Type  Always ‘S’ indicating Sequenced |
| 3 | 1 | Char | **OUCH Message Type**  **Always ‘S’ indicating System Event** |
| 4 | 8 | Unsigned Integer | Time Stamp |
| 12 | 1 | Char | Event Enumeration |

### Accepted Message

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset (bytes)** | **Size (bytes)** | **Type** | **Description** |
| 0 | 2 | Unsigned Integer | OUCH Message Length (excluding this field)  [expect 66] |
| 2 | 1 | Char | OUCH Packet Type  Always ‘S’ indicating Sequenced |
| 3 | 1 | Char | **OUCH Message Type**  **Always ‘A’ indicating Accepted** |
| 4 | 8 | Unsigned Integer | Time Stamp |
| 12 | 14 | Text | Order Token |
| 26 | 1 | Char | Side |
| 27 | 4 | Unsigned Integer | Shares |
| 31 | 8 | Text | Symbol |
| 39 | 4 | Unsigned Integer | Price (x 10,000) |
| 43 | 4 | Unsigned Integer | Time In Force |
| 47 | 4 | Text | Firm |
| 51 | 1 | Char | Display |
| 52 | 8 | Unsigned Integer | Order Reference Number |
| 60 | 1 | Char | Order Capacity |
| 61 | 1 | Char | Intermarket Sweep |
| 62 | 4 | Unsigned Integer | Minimum Quantity |
| 66 | 1 | Char | Cross Type |
| 67 | 1 | Char | Order State |

### Replaced Message

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset (bytes)** | **Size (bytes)** | **Type** | **Description** |
| 0 | 2 | Unsigned Integer | OUCH Message Length (excluding this field)  [expect 80] |
| 2 | 1 | Char | OUCH Packet Type  Always ‘S’ indicating Sequenced |
| 3 | 1 | Char | **OUCH Message Type**  **Always ‘U’ indicating Replaced** |
| 4 | 8 | Unsigned Integer | Time Stamp |
| 12 | 14 | Text | Order Token |
| 26 | 1 | Char | Side |
| 27 | 4 | Unsigned Integer | Shares |
| 31 | 8 | Text | Symbol |
| 39 | 4 | Unsigned Integer | Price (x 10,000) |
| 43 | 4 | Unsigned Integer | Time In Force |
| 47 | 4 | Text | Firm |
| 51 | 1 | Char | Display |
| 52 | 8 | Unsigned Integer | Order Reference Number |
| 60 | 1 | Char | Order Capacity |
| 61 | 1 | Char | Intermarket Sweep |
| 62 | 4 | Unsigned Integer | Minimum Quantity |
| 66 | 1 | Char | Cross Type |
| 67 | 1 | Char | Order State |
| 68 | 14 | Text | Previous Order Token |

### Executed Message

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset (bytes)** | **Size (bytes)** | **Type** | **Description** |
| 0 | 2 | Unsigned Integer | OUCH Message Length (excluding this field)  [expect 41] |
| 2 | 1 | Char | OUCH Packet Type  Always ‘S’ indicating Sequenced |
| 3 | 1 | Char | **OUCH Message Type**  **Always ‘E’ indicating Executed** |
| 4 | 8 | Unsigned Integer | Time Stamp |
| 12 | 14 | Text | Order Token |
| 26 | 4 | Unsigned Integer | Executed Shares |
| 30 | 4 | Unsigned Integer | Executed Price (x 10,000) |
| 34 | 1 | Char | Liquidity Flag |
| 35 | 8 | Unsigned Integer | Match Number |

### Canceled Message

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset (bytes)** | **Size (bytes)** | **Type** | **Description** |
| 0 | 2 | Unsigned Integer | OUCH Message Length (excluding this field)  [expect 29] |
| 2 | 1 | Char | OUCH Packet Type  Always ‘S’ indicating Sequenced |
| 3 | 1 | Char | **OUCH Message Type**  **Always ‘C’ indicating Canceled** |
| 4 | 8 | Unsigned Integer | Time Stamp |
| 12 | 14 | Text | Order Token |
| 26 | 4 | Unsigned Integer | Decrement Shares |
| 30 | 1 | Char | Reason |