

НИУ «МЭИ»

Отчет по первой части курсовой работы

по дисциплине

«Аппаратура потребителей спутниковых радионавигационных систем»

Выполнил: Капитонов А.И.

Вариант: 19

Москва, 2022

Этап 1. Обработка логов навигационного приемника

Цели:

1. Разработка программы, обрабатывающей входной файл in.txt и выводящей выходной файл out.txt, состоящий из таблицы эфемерид для спутника
2. Сравнить данные, полученные на выходе собственной программы и программы RTKNAVI.

Пункт 1.

Код программы написан в редакторе Qt Creator для упрощения работы по созданию Makefile и написанию кода программы. Для удобства части кода были разнесены на несколько файлов. На данном этапе реализован расчет всех целочисленных параметров спутника и с плавающей точкой.

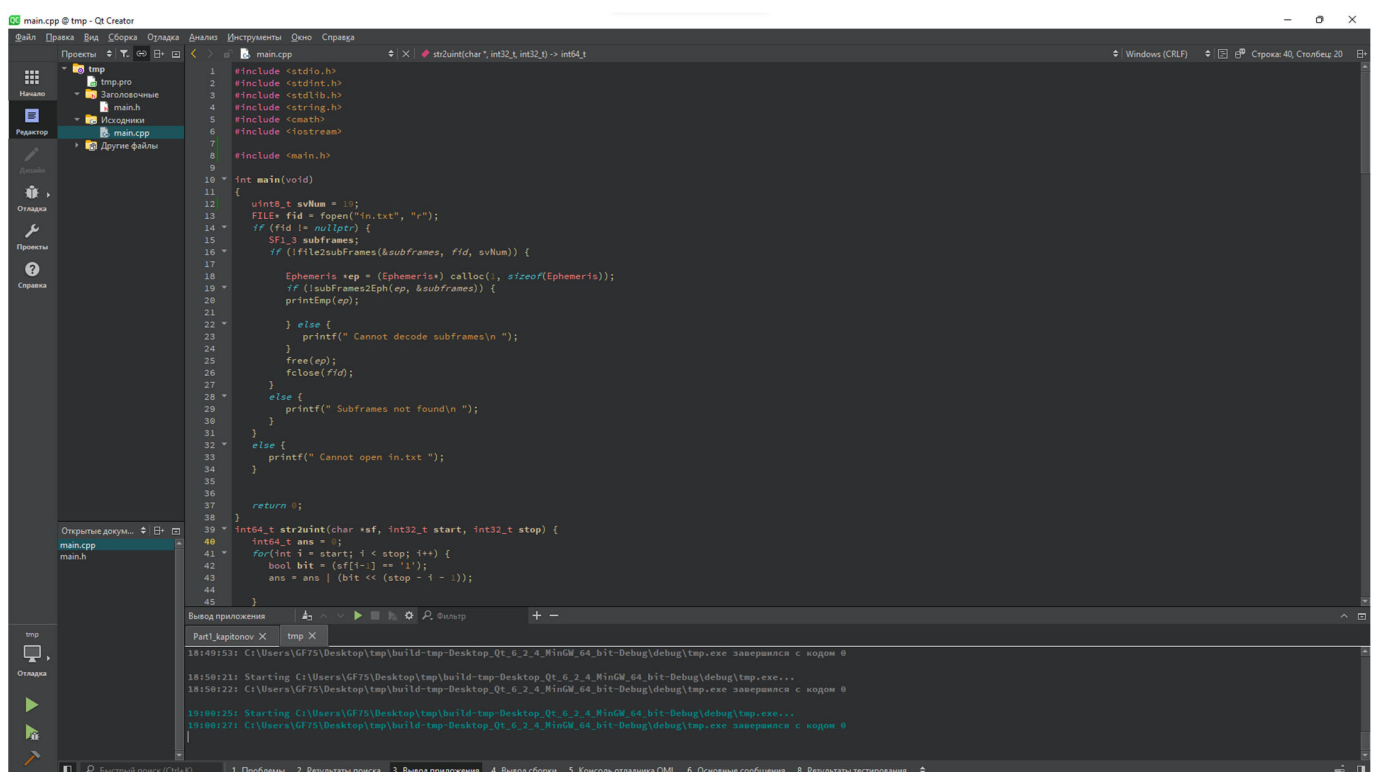


Рисунок 1 – интерфейс программы Qt и часть файла main.cpp

В файле main.cpp находятся вызовы основных функций:

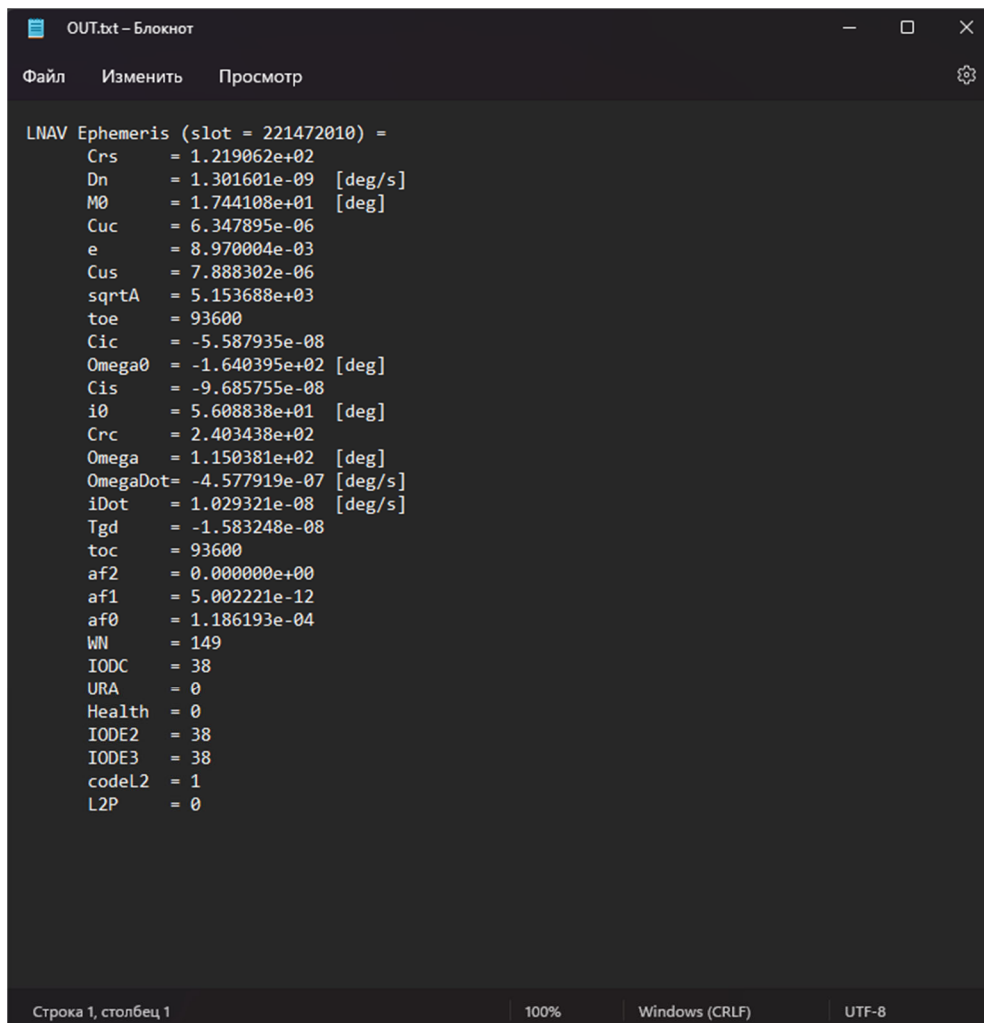
subFramesEph – расчет эфемерид из входного файла in.txt.

file2subFrames – функция парсинга строк входного файла на переменные,
необходимые для дальнейшей работы

printEph – вывод полученных данных в выходной файл OUT.txt и открытие этого
файла через notepad

RTKNAVI ver.2.4.2: RTK Monitor																				Close
Nav GPS	All	Current																		
e	IO (deg)	OMEGA0 (deg)	omega (deg)	M0 (deg)	deltan (deg/s)	OMEGA0dot (d	IDOT (deg/s)	afo (ns)	af1 (ns/s)	af2 (ns/s)	TGD (ns)	BGD5a(ns)	BGD5b(ns)	Cuc(rad)	Cus(rad)	Crc(m)	Crs(m)	Cic(rad)		
0.01137974	56.53057	-109.43551	50.51432	54.55674	2.4630E-07	-4.8124E-07	-6.7735E-09	432151.8	-0.0095	0.0000	5.1	0.0	0.0	7.7300E-07	1.1530E-06	3.7197E+02	1.1719E+01	-1.1735E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00389213	55.72485	-50.10616	54.89203	48.47481	2.3572E-07	-4.5681E-07	-1.7149E-08	-120342.7	-0.0172	0.0000	1.9	0.0	0.0	-6.625E-06	7.0781E-06	2.4825E+02	-1.2734E+02	-4.4703E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00728541	55.29557	-171.29829	5.90629	-115.58396	2.4679E-07	-4.5104E-07	9.9453E-09	-56090.3	-0.0015	0.0000	5.1	0.0	0.0	5.2750E-06	8.3819E-06	2.2300E+02	1.0591E+02	-5.5879E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00750119	55.71486	-50.26591	-144.08942	-1.69731	2.3953E-07	-4.6279E-07	-1.7312E-08	-319812.9	-0.0105	0.0000	2.3	0.0	0.0	-5.8059E-06	6.2771E-06	2.6297E+02	-1.1003E+02	2.2352E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00868590	55.59347	133.97619	71.43227	38.93650	2.4376E-07	-4.5196E-07	-3.1125E-08	-170663.5	-0.0057	0.0000	-12.6	0.0	0.0	3.0547E-07	9.8143E-06	1.9525E+02	8.4062E+00	-1.3411E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.01361771	56.16076	-166.60982	-85.86944	-120.70525	2.2469E-07	-4.3483E-07	1.1480E-08	574786.7	0.0050	0.0000	-11.2	0.0	0.0	6.9998E-06	7.8231E-06	2.3784E+02	1.3203E+02	1.7323E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00897000	56.08838	-164.03953	115.03807	17.44108	2.3429E-07	-4.5779E-07	1.0239E-08	118619.3	0.0050	0.0000	-15.8	0.0	0.0	6.3479E-06	7.8883E-06	2.4034E+02	1.2191E+02	-5.5879E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.02460024	54.96088	-114.64634	-57.65251	-110.18828	2.7215E-07	-4.7592E-07	5.8321E-09	157993.7	0.0007	0.0000	-10.2	0.0	0.0	2.5388E-06	1.4901E-06	3.5472E+02	3.8500E+01	-2.9244E		
0.01288441	55.09904	14.59397	-107.30072	-127.84517	2.4147E-07	-4.2956E-07	2.8281E-08	213230.0	0.0057	0.0000	-9.8	0.0	0.0	-3.8557E-07	9.9484E-06	1.8750E+02	-6.5938E+00	1.6950E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		
0.01045204	54.70826	71.24072	21.48581	50.45602	2.8242E-07	-4.7193E-07	1.8213E-09	-164610.3	-0.0018	0.0000	-13.5	0.0	0.0	6.8545E-07	5.2527E-06	2.7462E+02	1.0125E+01	-5.4017E		
0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E		

Рисунок 2 –Данные параметры с RTKNAVI



The image shows a Notepad window titled "OUT.txt - Блокнот". The menu bar includes "Файл", "Изменить", and "Просмотр". The text content is as follows:

```
LNAV Ephemeris (slot = 221472010) =  
  Crs      = 1.219062e+02  
  Dn       = 1.301601e-09 [deg/s]  
  M0       = 1.744108e+01 [deg]  
  Cus      = 6.347895e-06  
  e        = 8.970004e-03  
  Cus      = 7.888302e-06  
  sqrtA    = 5.153688e+03  
  toe      = 93600  
  Cic      = -5.587935e-08  
  Omega0   = -1.640395e+02 [deg]  
  Cis      = -9.685755e-08  
  i0       = 5.608838e+01 [deg]  
  Crc      = 2.403438e+02  
  Omega    = 1.150381e+02 [deg]  
  OmegaDot = -4.577919e-07 [deg/s]  
  iDot     = 1.029321e-08 [deg/s]  
  Tgd      = -1.583248e-08  
  toc      = 93600  
  af2      = 0.000000e+00  
  af1      = 5.002221e-12  
  af0      = 1.186193e-04  
  WN       = 149  
  IODC     = 38  
  URA      = 0  
  Health   = 0  
  IODE2    = 38  
  IODE3    = 38  
  codeL2   = 1  
  L2P      = 0
```

The status bar at the bottom shows "Строка 1, столбец 1", "100%", "Windows (CRLF)", and "UTF-8".

Рисунок 3 –Данные, полученные в ходе компиляции программы

Рассчитанные в коде программы данные и данные из программы RTKNAVI совпадают.

Приложение

Листинг программы для пункта 1

MAIN.CPP

```
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <string.h>
#include <cmath>
#include <iostream>

#include <main.h>

int main(void)
{
    uint8_t svNum = 19;
    FILE* fid = fopen("in.txt", "r");
    if (fid != nullptr) {
        SF1_3 subframes;
        if (!file2subFrames(&subframes, fid, svNum)) {

            Ephemeris *ep = (Ephemeris*) calloc(1, sizeof(Ephemeris));
            if (!subFrames2Eph(ep, &subframes)) {
                printEmp(ep);

            } else {
                printf(" Cannot decode subframes\n ");
            }
            free(ep);
            fclose(fid);
        }
        else {
            printf(" Subframes not found\n ");
        }
    }
    else {
        printf(" Cannot open in.txt ");
    }

    return 0;
}

int64_t str2uint(char *sf, int32_t start, int32_t stop) {
    int64_t ans = 0;
    for(int i = start; i < stop; i++) {
        bool bit = (sf[i-1] == '1');
        ans = ans | (bit << (stop - i - 1));
    }
    return ans;
}

int64_t str2int(uint64_t ans, int count) {

    int64_t Ians = 0;
    switch (count){
    case 8:
        if (bool((1<<7) & ans)){
            ans |= 0xFFFFFFFFFFFFFFF00;
            Ians = ~(ans - 1);
            return -Ians;
        }
    }
```

```

        break;
case 14:
    if (bool((1<<13) & ans)) {
        ans |= 0xFFFFFFFFFFFFC000;
        Ians = ~(ans - 1);
        return -Ians;}
    break;
case 16:
    if (bool((1 << 15) & ans)) {
        ans |= 0xFFFFFFFFFFFF0000;
        Ians = ~(ans - 1);
        return -Ians;}
    break;
case 22:
    if (bool((1 << 21) & ans)) {
        ans |= 0xFFFFFFFFFC00000;
        Ians = ~(ans - 1);
        return -Ians;}
    break;
case 24:
    if (bool((1 << 23) & ans)) {
        ans |= 0xFFFFFFFF000000;
        Ians = ~(ans - 1);
        return -Ians;}
    break;
case 32:
    if (bool((1 << 31) & ans)) {
        ans |= 0xFFFFFFFF00000000;
        Ians = ~(ans - 1);
        return -Ians;}
    break;
    }
    return ans;
}

int64_t str64uint(char *sf, int32_t start, int32_t stop, int32_t start2, int32_t
stop2){
    uint32_t ans = 0;
    for(int i = start; i < stop; i++) {
        ans = (ans | ((sf[i-1] == '1')? 1 : 0)) << 1;
    }
    for(int i = start2; i < stop2-1; i++) {

        ans = ans | ((sf[i-1] == '1')? 1 : 0);
        if (i < stop2-1) {
            ans = ans << 1;
        }
    }
    return ans;
}

int32_t subFrames2Eph(Ephemeris* ep, SF1_3* subframes) {
    ep->slot = subframes->slot;
    ep->Crs = str2int(str2uint(subframes->sf2,69,69+16),16)*pow(2, -5);
    ep->Dn = str2int(str2uint(subframes->sf2, 91, 91+16), 16)*pow(2, -43);
    ep->M0 = str2int(str64uint(subframes->sf2,107, 107+8, 121, 121+24),32)*pow(2, -
31)*180;
    ep->Cuc = str2int(str2uint(subframes->sf2,151,151+16),16)*pow(2, -29);
    ep->e = str64uint(subframes->sf2,167, 167+8, 181, 181+24) * pow(2, -33);
    ep->Cus = str2int(str2uint(subframes->sf2,211,211+16),16)*pow(2, -29);
    ep->sqrta = str64uint(subframes->sf2,227, 227+8, 241, 241+24) * pow(2, -19);
    ep->toe = str2uint(subframes->sf2, 271, 287)*pow(2,4);
    ep->Cic = str2int(str2uint(subframes->sf3,61,61+16),16)*pow(2, -29);

```

```

    ep->Omega0 = str2int(str64uint(subframes->sf3,77, 77+8, 91, 91+24),32)*pow(2, -
31)*180;
    ep->Cis = str2int(str2uint(subframes->sf3,121,121+16),16)*pow(2, -29);
    ep->i0 = str2int(str64uint(subframes->sf3,137, 137+8, 151, 151+24),32)*pow(2, -
31)*180;
    ep->Crc = str2int(str2uint(subframes->sf3,181,181+16),16)*pow(2, -5);
    ep->Omega = str2int(str64uint(subframes->sf3,197, 197+8, 211, 211+24),32)*pow(2, -
31)*180;
    ep->OmegaDot = str2int(str2uint(subframes->sf3,241,241+24),24)*pow(2, -43)*180;
    ep->iDot = str2int(str2uint(subframes->sf3,279,279+14),14)*pow(2, -43)*180;
    ep->Tgd = str2int(str2uint(subframes->sf1,197,197+8),8)*pow(2, -31);
    ep->toc = str2int(str2uint(subframes->sf1,219,219+16),16)*pow(2, 4);
    ep->af2 = str2int(str2uint(subframes->sf1,241,241+8),8)*pow(2, -55);
    ep->af1 = str2int(str2uint(subframes->sf1,249,249+16),16)*pow(2, -43);
    ep->af0 = str2int(str2uint(subframes->sf1,271,271+22),22)*pow(2, -31);
    ep->WN = str2uint(subframes->sf1, 61, 71);
    ep->IODC = str64uint(subframes->sf1,83, 83+2, 211, 211+8);
    ep->URA = str2uint(subframes->sf1, 73, 77);
    ep->Health = str2uint(subframes->sf1, 73, 73+6);
    ep->IODE2 = str2uint(subframes->sf2, 61, 69);
    ep->IODE3 = str2uint(subframes->sf3, 271, 271+8);
    ep->codeL2 = str2uint(subframes->sf1, 71, 73);
    ep->L2P = str2uint(subframes->sf1, 90, 91);
    return 0;
}

```

```

int32_t file2subFrames(SF1_3* sf, FILE* fid, uint8_t svNum){
    int32_t sth1, sth2, sth3, sth4, sth5;
    char str_OR[8];
    char str_GPSL1CA[12];
    char str_reh[8];
    char str[1000];
    uint32_t svStr;
    uint32_t slot;
    int32_t subFrameNum;

    uint32_t slot_SF1 = 0;
    uint32_t slot_SF2 = 0;
    uint32_t slot_SF3 = 0;
    int32_t readres = 0;

    while(readres != EOF)
    {
        svStr = 0;
        readres = fscanf( fid, "%d %d %d %s %s %s %u\t %u %d %d %d %s", &sth1, &sth2,
&sth3, str_OR, str_GPSL1CA, str_reh, &svStr, &slot, &sth4, &sth5, &subFrameNum, str);
        if (( svStr == svNum ) && (slot >= (604800/6))) {
            if ( subFrameNum == 1 ) {
                slot_SF1 = slot;
                strncpy(sf->sf1, str, sizeof(sf->sf1));
            }
            else if (subFrameNum == 2) {
                slot_SF2 = slot;
                strncpy(sf->sf2, str, sizeof(sf->sf2));
            }
            else if (subFrameNum == 3) {
                slot_SF3 = slot;
                strncpy(sf->sf3, str, sizeof(sf->sf3));
            }
            if ((slot_SF1 + 1 == slot_SF2) && (slot_SF2 + 1 == slot_SF3)) {
                sf->slot = slot_SF1;
                return 0;
            }
        }
    }
}

```

```

    }
    return 1;
}

void printEmp(Ephemeris* ep)
{
    FILE* OUT;
    OUT = fopen("OUT.txt", "w");
    fprintf(OUT, "LNAV Ephemeris (slot = %u) = \n", ep->slot );
    fprintf(OUT, "\tCrs      = %e \n", ep->Crs );
    fprintf(OUT, "\tDn      = %e \t[deg/s] \n", ep->Dn );
    fprintf(OUT, "\tM0      = %e \t[deg] \n", ep->M0 );
    fprintf(OUT, "\tCuc      = %e \n", ep->Cuc );
    fprintf(OUT, "\te       = %e \n", ep->e );
    fprintf(OUT, "\tCus      = %e \n", ep->Cus );
    fprintf(OUT, "\tsqrtA    = %e \n", ep->sqrtA );
    fprintf(OUT, "\ttoe      = %u \n", ep->toe );
    fprintf(OUT, "\tCic      = %e \n", ep->Cic );
    fprintf(OUT, "\tOmega0    = %e [deg] \n", ep->Omega0 );
    fprintf(OUT, "\tCis      = %e \n", ep->Cis );
    fprintf(OUT, "\ti0       = %e \t[deg] \n", ep->i0 );
    fprintf(OUT, "\tCrc      = %e \n", ep->Crc );
    fprintf(OUT, "\tOmega     = %e \t[deg] \n", ep->Omega );
    fprintf(OUT, "\tOmegaDot  = %e [deg/s] \n", ep->OmegaDot );
    fprintf(OUT, "\tiDot     = %e \t[deg/s] \n", ep->iDot );
    fprintf(OUT, "\tTgd      = %e \n", ep->Tgd );
    fprintf(OUT, "\ttoc      = %u \n", ep->toc );
    fprintf(OUT, "\taf2      = %e \n", ep->af2 );
    fprintf(OUT, "\taf1      = %e \n", ep->af1 );
    fprintf(OUT, "\taf0      = %e \n", ep->af0 );
    fprintf(OUT, "\tWN       = %u \n", ep->WN );
    fprintf(OUT, "\tIODC     = %u \n", ep->IODC );
    fprintf(OUT, "\tURA      = %u \n", ep->URA );
    fprintf(OUT, "\tHealth    = %u \n", ep->Health );
    fprintf(OUT, "\tIODE2     = %u \n", ep->IODE2 );
    fprintf(OUT, "\tIODE3     = %u \n", ep->IODE3 );
    fprintf(OUT, "\tcodeL2    = %u \n", ep->codeL2 );
    fprintf(OUT, "\tL2P      = %u \n", ep->L2P );
    fclose(OUT);
    system("OUT.txt");
}

```

MAIN.H

```

#ifndef MAIN_H
#define MAIN_H
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <string.h>
#define LENGTH 301

struct Ephemeris {
    double   Crs;
    double   Dn;
    double   M0;
    double   Cuc;
    double   e;
    double   Cus;
    double   sqrtA;
    uint32_t toe;
    double   Cic;
    double   Omega0;
    double   Cis;
    double   i0;

```



```

double    Crc;
double    Omega;
double    OmegaDot;
double    iDot;
double    Tgd;
uint32_t  toc;
double    af2;
double    af1;
double    af0;
uint32_t  WN;
uint16_t  IODC;
uint8_t   URA;
uint8_t   Health;
uint16_t  IODE2;
uint16_t  IODE3;
bool      codeL2;
bool      L2P;
uint32_t  slot;
};

struct SF1_3 {
    uint32_t slot;
    char sf1[LENGTH];
    char sf2[LENGTH];
    char sf3[LENGTH];
};

void printEmp(Ephemeris* ep);
int32_t file2subFrames(SF1_3* sf, FILE* fid, uint8_t svNum);
int32_t subFrames2Eph(Ephemeris* ep, SF1_3* subframes);
int64_t str2uint(char *sf, int32_t start, int32_t stop);
int64_t str2int(uint64_t ans, int count);
int64_t str64uint(char *sf, int32_t start, int32_t stop, int32_t start2, int32_t
stop2);

#endif // MAIN_H

```