```
int Reverse(char date[], char angle[])
{
   /**************************************************************************/

   // ディレクトリ文字列作成
   char directoryname[100];
   char directoryname_csv[100];
   char directoryname_dat[100];
   char directoryname_png[100];

   /**************************************************************************/

   // 元ディレクトリの作成
   sprintf(directoryname, "../Result/%s", date);
   mkdir(directoryname, mode);

   /**************************************************************************/

   // ディレクトリの作成
   sprintf(directoryname, "../Result/%s/010_Reverse", date);
   mkdir(directoryname, mode);

   /**************************************************************************/

   // ディレクトリの作成
   sprintf(directoryname_dat, "../Result/%s/010_Reverse/dat", date);
   sprintf(directoryname_csv, "../Result/%s/010_Reverse/csv", date);
   sprintf(directoryname_png, "../Result/%s/010_Reverse/png", date);

   mkdir(directoryname_dat, mode);
   mkdir(directoryname_csv, mode);
   mkdir(directoryname_png, mode);

   /**************************************************************************/

   // ファイルの指定
   char filename_read[100];
   char filename_csv[100];
   char filename_dat[100];

   sprintf(filename_read, "../Data/%s/%s.csv", date, angle);
   sprintf(filename_csv, "../Result/%s/010_Reverse/csv/%s.csv", date, angle);
   sprintf(filename_dat, "../Result/%s/010_Reverse/dat/%s.dat", date, angle);

   /**************************************************************************/

   // 変数宣言
   int i, j;
   int datalength = 0;
   int ch = 3;
   double value[5000][ch];
   double ch0, ch1, ch2; // ch0:drag, ch1:lift, ch2:load-cell

   double time = 0;

   // 配列の初期化

   for (i = 0; i < 5000; i++)
   {
      for (j = 0; j < 3; j++)
      {
         value[i][j] = 0;
      }
   }

   // printf("check");

   // ファイルの読み込み
   fp = fopen(filename_read, "r");
   if (fp == NULL)
   {
      printf("01\t[%s]\tno data file\n", angle);
      return 1;
   }

   i = 0;

   while ((fscanf(fp, "%lf, %lf, %lf", &ch0, &ch1, &ch2)) != EOF)
   {
      // printf("%.3f, %.3f, %.3f\n", ch0, ch1, ch2);
      value[i][0] = ch0;
      value[i][1] = ch1;
      value[i][2] = ch2;
      i = i + 1;
   }

   fclose(fp);

   datalength = i;

   // 計算

   for (i = 0; i < datalength; i++)
   {
      value[i][2] = -1 * value[i][2];
   }

   fp_csv = fopen(filename_csv, "w");
   fp_dat = fopen(filename_dat, "w");

   for (i = 0; i < datalength; i++)
   {
      time = i * 0.2;
   fprintf(fp_csv, "%.3f,%.3f,%.3f\n", value[i][0], value[i][1], value[i][2]);
   fprintf(fp_dat, "%d\t%.3f\t%.3f\t%.3f\t%.1f\n", i, value[i][0], value[i][1], value[i][2], time);
}

fclose(fp_csv);
fclose(fp_dat);

/********************************************************************/
// Gnuplot //

// ディレクトリの作成
char directoryname_png_1[100];
char directoryname_png_2[100];
char directoryname_png_3[100];

sprintf(directoryname_png_1, "../Result/%s/010_Reverse/png/Loadcell", date);
sprintf(directoryname_png_2, "../Result/%s/010_Reverse/png/Strainsensors", date);
sprintf(directoryname_png_3, "../Result/%s/010_Reverse/png/Allsensors", date);

mkdir(directoryname_png_1, mode);
mkdir(directoryname_png_2, mode);
mkdir(directoryname_png_3, mode);

/********************************************************************/

// filename
char filename_png_1[100];
char filename_png_2[100];
char filename_png_3[100];

sprintf(filename_dat, "../Result/%s/010_Reverse/dat/%s.dat", date, angle);
sprintf(filename_png_1, "../Result/%s/010_Reverse/png/Loadcell/%s.png", date, angle);
sprintf(filename_png_2, "../Result/%s/010_Reverse/png/Strainsensors/%s.png", date, angle);
sprintf(filename_png_3, "../Result/%s/010_Reverse/png/Allsensors/%s.png", date, angle);

/********************************************************************/

// range x
int x_min = 0;
int x_max = 600;

// range y
double y_min = -1;
double y_max = 1;

// range y (loadcell)
double y_min_loadcell = -0.5;
double y_max_loadcell = 2.5;

// label
const char *xxlabel = "Time [s]";
const char *yylabel = "Output voltage [V]";
char label[100];

double size;

double angle_2 = 0;

angle_2 = atoi(angle);
angle_2 = angle_2;
sprintf(label, "%0.f", angle_2);

// size
size = 1;

/********************************************************************/

if ((gp = popen("gnuplot", "w")) == NULL)
{
   printf("gnuplot is not here!\n");
   exit(0); // が無い場合、異常ある場合は終了 gnuplot
}

fprintf(gp, "set terminal pngcairo enhanced font ’Times New Roman,15’ \n");
fprintf(gp, "set output ’%s’\n", filename_png_1);
// fprintf(gp, "set multiplot\n");
fprintf(gp, "unset key\n");
fprintf(gp, "set term pngcairo size 1280, 960 font ’,27’\n");
// fprintf(gp, "set size ratio %.3f\n", size);

fprintf(gp, "set lmargin screen 0.10\n");
fprintf(gp, "set rmargin screen 0.90\n");
fprintf(gp, "set tmargin screen 0.90\n");
fprintf(gp, "set bmargin screen 0.15\n");

fprintf(gp, "set xrange [%d:%d]\n", x_min, x_max);
fprintf(gp, "set xlabel ’%s’offset 0.0,0\n", xxlabel);
fprintf(gp, "set yrange [%.3f:%.3f]\n", y_min_loadcell, y_max_loadcell);
fprintf(gp, "set ylabel ’%s’offset 1.0,0.0\n", yylabel);
fprintf(gp, "set title ’Load cell : %s [deg]’\n", label);

// fprintf(gp, "set samples 10000\n");
fprintf(gp, "plot ’%s’ using 5:4 with lines lc ’black’ title ’loadcell’\n", filename_dat);
fflush(gp); // Clean up data

/********************************************************************/

fprintf(gp, "set terminal pngcairo enhanced font ’Times New Roman,15’ \n");
fprintf(gp, "set output ’%s’\n", filename_png_2);
// fprintf(gp, "set multiplot\n");
fprintf(gp, "set key left top\n");
fprintf(gp, "set key font ’,22’\n");
fprintf(gp, "set term pngcairo size 1280, 960 font ’,27’\n");
```

```
    // fprintf(gp, "set size ratio %.3f\n", size);

    fprintf(gp, "set lmargin screen 0.10\n");
    fprintf(gp, "set rmargin screen 0.90\n");
    fprintf(gp, "set tmargin screen 0.90\n");
    fprintf(gp, "set bmargin screen 0.15\n");

    fprintf(gp, "set xrange [%d:%d]\n", x_min, x_max);
    fprintf(gp, "set xlabel ’%s’offset 0.0,0\n", xxlabel);
    fprintf(gp, "set yrange [%.3f:%.3f]\n", y_min, y_max);
    fprintf(gp, "set ylabel ’%s’offset 1.0,0.0\n", yylabel);
    fprintf(gp, "set title ’Strain sensors : %s [deg]’\n", label);

    // fprintf(gp, "set samples 10000\n");
    fprintf(gp, "plot ’%s’ using 5:2 with lines lc ’blue’ title ’Drag’, ’%s’ using 5:3 with lines lc ’red’
         title ’Lift’\n", filename_dat, filename_dat);
    fflush(gp); // Clean up data

    /**************************************************************************/

    fprintf(gp, "set terminal pngcairo enhanced font ’Times New Roman,15’ \n");
    fprintf(gp, "set output ’%s’\n", filename_png_3);
    // fprintf(gp, "set multiplot\n");
    fprintf(gp, "set key left top\n");
    fprintf(gp, "set key font ’,22’\n");
    fprintf(gp, "set term pngcairo size 1280, 960 font ’,27’\n");
    // fprintf(gp, "set size ratio %.3f\n", size);

    fprintf(gp, "set lmargin screen 0.10\n");
    fprintf(gp, "set rmargin screen 0.90\n");
    fprintf(gp, "set tmargin screen 0.90\n");
    fprintf(gp, "set bmargin screen 0.15\n");

    fprintf(gp, "set xrange [%d:%d]\n", x_min, x_max);
    fprintf(gp, "set xlabel ’%s’offset 0.0,0\n", xxlabel);
    fprintf(gp, "set yrange [%.3f:%.3f]\n", y_min, y_max_loadcell);
    fprintf(gp, "set ylabel ’%s’offset 1.0,0.0\n", yylabel);
    fprintf(gp, "set title ’All sensors : %s [deg]’\n", label);

    // fprintf(gp, "set samples 10000\n");
    fprintf(gp, "plot ’%s’ using 5:4 with lines lc ’black’ title ’loadcell’, ’%s’ using 5:2 with lines lc ’
         blue’ title ’Drag’, ’%s’ using 5:3 with lines lc ’red’ title ’Lift’\n", filename_dat, filename_dat,
          filename_dat);
    fflush(gp); // Clean up data

    fprintf(gp, "exit\n"); // Quit gnuplot

    pclose(gp);

    printf("01\t[%s]\tsuccess\n", angle);
}
```