```
/*  -*- Last-Edit:  Fri Jan 29 11:13:27 1993 by Tarak S. Goradia; -*- */
/* $Log: tcas.c,v $
 * Revision 1.2  1993/03/12  19:29:50  foster
 * Correct logic bug which didn't allow output of 2 - hf
 * */

#include <stdio.h>

#define OLEV       600          /* in feets/minute */
#define MAXALTDIFF 600          /* max altitude difference in feet */
#define MINSEP     300          /* min separation in feet */
#define NOZCROSS   100          /* in feet */
                                /* variables */

typedef int bool;

int Cur_Vertical_Sep;
bool High_Confidence;
bool Two_of_Three_Reports_Valid;

int Own_Tracked_Alt;
int Own_Tracked_Alt_Rate;
int Other_Tracked_Alt;

int Alt_Layer_Value;            /* 0, 1, 2, 3 */
int Positive_RA_Alt_Thresh[4];

int Up_Separation;
int Down_Separation;

                                /* state variables */
int Other_RAC;                  /* NO_INTENT, DO_NOT_CLIMB, DO_NOT_DESCEND */
#define NO_INTENT 0
#define DO_NOT_CLIMB 1
#define DO_NOT_DESCEND 2

int Other_Capability;           /* TCAS_TA, OTHER */
#define TCAS_TA 1
#define OTHER 2

int Climb_Inhibit;              /* true/false */

#define UNRESOLVED 0
#define UPWARD_RA 1
#define DOWNWARD_RA 2

void initialize()
{
    Positive_RA_Alt_Thresh[0] = 400;
    Positive_RA_Alt_Thresh[1] = 500;
    Positive_RA_Alt_Thresh[2] = 640;
    Positive_RA_Alt_Thresh[3] = 740;
}

int ALIM ()
{
 return Positive_RA_Alt_Thresh[Alt_Layer_Value];
}

int Inhibit_Biased_Climb ()
{
    return (Climb_Inhibit ? Up_Separation + NOZCROSS : Up_Separation);
}

bool Non_Crossing_Biased_Climb()
{
    int upward_preferred;
    int upward_crossing_situation;
    bool result;

    upward_preferred = Inhibit_Biased_Climb() > Down_Separation;
    if (upward_preferred)
```



```
    {
        result = !(Own_Below_Threat()) || ((Own_Below_Threat()) && (!(Down_Separ
ation >= ALIM())));
    }
    else
    {
        result = Own_Above_Threat() && (Cur_Vertical_Sep >= MINSEP) && (Up_Separ
ation >= ALIM());
    }
    return result;
}

bool Non_Crossing_Biased_Descend()
{
    int upward_preferred;
    int upward_crossing_situation;
    bool result;

    upward_preferred = Inhibit_Biased_Climb() > Down_Separation;
    if (upward_preferred)
    {
        result = Own_Below_Threat() && (Cur_Vertical_Sep >= MINSEP) && (Down_Sep
aration >= ALIM());
    }
    else
    {
        result = !(Own_Above_Threat()) || ((Own_Above_Threat()) && (Up_Separatio
n >= ALIM()));
    }
    return result;
}

bool Own_Below_Threat()
{
    return (Own_Tracked_Alt < Other_Tracked_Alt);
}

bool Own_Above_Threat()
{
    return (Other_Tracked_Alt < Own_Tracked_Alt);
}

int alt_sep_test()
{
    bool enabled, tcas_equipped, intent_not_known;
    bool need_upward_RA, need_downward_RA;
    int alt_sep;

    enabled = High_Confidence && (Own_Tracked_Alt_Rate <= OLEV) && (Cur_Vertical
_Sep > MAXALTDIFF);
    tcas_equipped = Other_Capability == TCAS_TA;
    intent_not_known = Two_of_Three_Reports_Valid && Other_RAC == NO_INTENT;

    alt_sep = UNRESOLVED;

    if (enabled && ((tcas_equipped && intent_not_known) || !tcas_equipped))
    {
        need_upward_RA = Non_Crossing_Biased_Climb() && Own_Below_Threat();
        need_downward_RA = Non_Crossing_Biased_Descend() && Own_Above_Threat();
        if (need_upward_RA && need_downward_RA)
        /* unreachable: requires Own_Below_Threat and Own_Above_Threat
           to both be true - that requires Own_Tracked_Alt < Other_Tracked_Alt
           and Other_Tracked_Alt < Own_Tracked_Alt, which isn't possible */
            alt_sep = UNRESOLVED;
        else if (need_upward_RA)
            alt_sep = UPWARD_RA;
        else if (need_downward_RA)
            alt_sep = DOWNWARD_RA;
        else
            alt_sep = UNRESOLVED;
    }

    return alt_sep;
```

```
142 }
143
144 main(argc, argv)
145 int argc;
146 char *argv[];
147 {
148     if(argc < 13)
149     {
150         fprintf(stdout, "Error: Command line arguments are\n");
151         fprintf(stdout, "Cur_Vertical_Sep, High_Confidence, Two_of_Three_Reports_Valid\n");
152         fprintf(stdout, "Own_Tracked_Alt, Own_Tracked_Alt_Rate, Other_Tracked_Alt\n");
153         fprintf(stdout, "Alt_Layer_Value, Up_Separation, Down_Separation\n");
154         fprintf(stdout, "Other_RAC, Other_Capability, Climb_Inhibit\n");
155         exit(1);
156     }
157     initialize();
158     Cur_Vertical_Sep = atoi(argv[1]);
159     High_Confidence = atoi(argv[2]);
160     Two_of_Three_Reports_Valid = atoi(argv[3]);
161     Own_Tracked_Alt = atoi(argv[4]);
162     Own_Tracked_Alt_Rate = atoi(argv[5]);
163     Other_Tracked_Alt = atoi(argv[6]);
164     Alt_Layer_Value = atoi(argv[7]);
165     Up_Separation = atoi(argv[8]);
166     Down_Separation = atoi(argv[9]);
167     Other_RAC = atoi(argv[10]);
168     Other_Capability = atoi(argv[11]);
169     Climb_Inhibit = atoi(argv[12]);
170
171     fprintf(stdout, "%d\n", alt_sep_test());
172     exit(0);
173 }
```