

```

//-----Array for pressure calculation, unit 100 * mBar-----
long P_sol[12] = {873,1228,1706,2339,3169,4246,5627,7381,9590,12340,15750,19930};

//-----Arrays for concentration calculations, unit 100 * %-----
long T_sol_20[12] = {4585,4026,3010,0,0,0,0,0,0,0,0,0};
long T_sol_25[12] = {4953,4542,3976,2957,0,0,0,0,0,0,0,0};
long T_sol_30[12] = {5257,4914,4498,3927,2906,0,0,0,0,0,0,0};
long T_sol_35[12] = {5528,5219,4874,4455,3878,2858,0,0,0,0,0,0};
long T_sol_40[12] = {5781,5491,5182,4834,4412,3831,2813,0,0,0,0,0};
long T_sol_45[12] = {6025,5743,5454,5144,4795,4370,3784,0,0,0,0,0};
long T_sol_50[12] = {6269,5986,5705,5417,5107,4756,4328,0,0,0,0,0};
long T_sol_55[12] = {6514,6226,5947,5668,5380,5070,4718,0,0,0,0,0};
long T_sol_60[12] = {6764,6467,6184,5909,5632,5344,5034,4680,4247,3652,2653,0};
long T_sol_65[12] = {7014,6710,6422,6144,5871,5596,5309,4997,4642,4207,3610,2617};
long T_sol_70[12] = {7256,6955,6660,6378,6104,5834,5560,5273,4961,4604,4167,3569};
long T_sol_75[12] = {7469,7193,6898,6612,6335,6066,5798,5525,5238,4925,4567,4128};
long T_sol_80[12] = {7649,7408,7132,6845,6565,6294,6028,5762,5490,5203,4890,4531};
long T_sol_85[12] = {7799,7591,7348,7074,6794,6520,6254,5991,5727,5455,5169,4855};
long T_sol_90[12] = {7889,7751,7534,7289,7018,6745,6477,6215,5954,5692,5421,5134};
long T_sol_95[12] = {7943,7860,7698,7479,7232,6965,6698,6435,6176,5918,5657,5387};
long T_sol_100[12] = {7980,7924,7825,7643,7424,7176,6914,6652,6394,6139,5883,5623};
long T_sol_105[12] = {8000,7966,7902,7785,7590,7370,7123,6865,6609,6355,6102,5848};
long T_sol_110[12] = {8000,7998,7950,7876,7738,7538,7316,7071,6818,6566,6316,6066};
long T_sol_115[12] = {8000,8000,7985,7933,7846,7688,7488,7264,7021,6773,6525,6278};
long T_sol_120[12] = {8000,8000,8000,7972,7913,7812,7638,7438,7213,6973,6729,6486};

//-----Output variables-----
float P_solution_sep = 0;
float Concentr_sep = 0;
float P_solution_evap = 0;
float Concentr_absorb = 0;

void setup()
{
  //-----nothing here-----
}

void loop()
{
  //
  // read and convert all the temperatures
  //
  //
  //-----claculate pressures and concentrations-----
  P_solution_sep = pressure(T_sep_out_gas);
  P_solution_evap = pressure(T_evaporator);
  Concentr_sep = concentration(T_sep_out_gas, T_sep_out_liq);
  Concentr_absorb = concentration(T_evaporator, T_absorb_out);
  //
}

//----- pressure -----
float pressure (float temp1) {
  int index1;
  float fraction1;
  float p_sat;
  if ((temp1 < 5) || (temp1 > 60)) { // out of range test
    p_sat = 0;
  }
  else {
    index1 = int(temp1/5)-1; //index i array starts with 0, and index of 5 dgrC must be 0
    fraction1 = (temp1 - ((index1 + 1) * 5)) * 20; //output in % of one step
    p_sat = (P_sol[index1] + (P_sol[index1+1] - P_sol[index1]) * fraction1/100)/100;
  }
  return p_sat;
}

//-----concentration-----
float concentration (float temp1, float temp2) {
  int index1;
  int index2;
  float fraction1;
  float fraction2;
  float interrim1;
  float interrim2;
  float p_sat;
  float conc;

  if ((temp1 < 5) || (temp1 > 60)) { // out of range test
    p_sat = 0;
  }
  else {
    index1 = int(temp1 / 5) - 1; //index i array starts with 0, and index of 5 dgrC must be 0
    fraction1 = (temp1 - ((index1 + 1) * 5)) * 20; //output in % of one step
    p_sat = (P_sol[index1] + (P_sol[index1+1] - P_sol[index1]) * fraction1 / 100)/100;
  }

  if ((temp2 < 20) || (temp2 > 120) || (p_sat == 0)) { //out of range test
    conc = 0;
  }
  else {
    index2 = int(temp2 / 5) - 3; //here we start indexing with 1 for 20 dgrC (and not 0 as in arrays)
    fraction2 = (temp2 - ((index2 + 3) * 5)) * 20;
  }
}

```

```

switch (index2) {
case 1:
    interr1 = (T_sol_20[index1] + ((T_sol_20[index1+1] - T_sol_20[index1]) * fraction1)/100);
    interr2 = (T_sol_25[index1] + ((T_sol_25[index1+1] - T_sol_25[index1]) * fraction1)/100);
    break;
case 2:
    interr1 = (T_sol_25[index1] + ((T_sol_25[index1+1] - T_sol_25[index1]) * fraction1)/100);
    interr2 = (T_sol_30[index1] + ((T_sol_30[index1+1] - T_sol_30[index1]) * fraction1)/100);
    break;
case 3:
    interr1 = (T_sol_30[index1] + ((T_sol_30[index1+1] - T_sol_30[index1]) * fraction1)/100);
    interr2 = (T_sol_35[index1] + ((T_sol_35[index1+1] - T_sol_35[index1]) * fraction1)/100);
    break;
case 4:
    interr1 = (T_sol_35[index1] + ((T_sol_35[index1+1] - T_sol_35[index1]) * fraction1)/100);
    interr2 = (T_sol_40[index1] + ((T_sol_40[index1+1] - T_sol_40[index1]) * fraction1)/100);
    break;
case 5:
    interr1 = (T_sol_40[index1] + ((T_sol_40[index1+1] - T_sol_40[index1]) * fraction1)/100);
    interr2 = (T_sol_45[index1] + ((T_sol_45[index1+1] - T_sol_45[index1]) * fraction1)/100);
    break;
case 6:
    interr1 = (T_sol_45[index1] + ((T_sol_45[index1+1] - T_sol_45[index1]) * fraction1)/100);
    interr2 = (T_sol_50[index1] + ((T_sol_50[index1+1] - T_sol_50[index1]) * fraction1)/100);
    break;
case 7:
    interr1 = (T_sol_50[index1] + ((T_sol_50[index1+1] - T_sol_50[index1]) * fraction1)/100);
    interr2 = (T_sol_55[index1] + ((T_sol_55[index1+1] - T_sol_55[index1]) * fraction1)/100);
    break;
case 8:
    interr1 = (T_sol_55[index1] + ((T_sol_55[index1+1] - T_sol_55[index1]) * fraction1)/100);
    interr2 = (T_sol_60[index1] + ((T_sol_60[index1+1] - T_sol_60[index1]) * fraction1)/100);
    break;
case 9:
    interr1 = (T_sol_60[index1] + ((T_sol_60[index1+1] - T_sol_60[index1]) * fraction1)/100);
    interr2 = (T_sol_65[index1] + ((T_sol_65[index1+1] - T_sol_65[index1]) * fraction1)/100);
    break;
case 10:
    interr1 = (T_sol_65[index1] + ((T_sol_65[index1+1] - T_sol_65[index1]) * fraction1)/100);
    interr2 = (T_sol_70[index1] + ((T_sol_70[index1+1] - T_sol_70[index1]) * fraction1)/100);
    break;
case 11:
    interr1 = (T_sol_70[index1] + ((T_sol_70[index1+1] - T_sol_70[index1]) * fraction1)/100);
    interr2 = (T_sol_75[index1] + ((T_sol_75[index1+1] - T_sol_75[index1]) * fraction1)/100);
    break;
case 12:
    interr1 = (T_sol_75[index1] + ((T_sol_75[index1+1] - T_sol_75[index1]) * fraction1)/100);
    interr2 = (T_sol_80[index1] + ((T_sol_80[index1+1] - T_sol_80[index1]) * fraction1)/100);
    break;
case 13:
    interr1 = (T_sol_80[index1] + ((T_sol_80[index1+1] - T_sol_80[index1]) * fraction1)/100);
    interr2 = (T_sol_85[index1] + ((T_sol_85[index1+1] - T_sol_85[index1]) * fraction1)/100);
    break;
case 14:
    interr1 = (T_sol_85[index1] + ((T_sol_85[index1+1] - T_sol_85[index1]) * fraction1)/100);
    interr2 = (T_sol_90[index1] + ((T_sol_90[index1+1] - T_sol_90[index1]) * fraction1)/100);
    break;
case 15:
    interr1 = (T_sol_90[index1] + ((T_sol_90[index1+1] - T_sol_90[index1]) * fraction1)/100);
    interr2 = (T_sol_95[index1] + ((T_sol_95[index1+1] - T_sol_95[index1]) * fraction1)/100);
    break;
case 16:
    interr1 = (T_sol_95[index1] + ((T_sol_95[index1+1] - T_sol_95[index1]) * fraction1)/100);
    interr2 = (T_sol_100[index1] + ((T_sol_100[index1+1] - T_sol_100[index1]) * fraction1)/100);
    break;
case 17:
    interr1 = (T_sol_100[index1] + ((T_sol_100[index1+1] - T_sol_100[index1]) * fraction1)/100);
    interr2 = (T_sol_105[index1] + ((T_sol_105[index1+1] - T_sol_105[index1]) * fraction1)/100);
    break;
case 18:
    interr1 = (T_sol_105[index1] + ((T_sol_105[index1+1] - T_sol_105[index1]) * fraction1)/100);
    interr2 = (T_sol_110[index1] + ((T_sol_110[index1+1] - T_sol_110[index1]) * fraction1)/100);
    break;
case 19:
    interr1 = (T_sol_110[index1] + ((T_sol_110[index1+1] - T_sol_110[index1]) * fraction1)/100);
    interr2 = (T_sol_115[index1] + ((T_sol_115[index1+1] - T_sol_115[index1]) * fraction1)/100);
    break;
case 20:
    interr1 = (T_sol_115[index1] + ((T_sol_115[index1+1] - T_sol_115[index1]) * fraction1)/100);
    interr2 = (T_sol_120[index1] + ((T_sol_120[index1+1] - T_sol_120[index1]) * fraction1)/100);
    break;
case 21:
    interr1 = (T_sol_120[index1] + ((T_sol_120[index1+1] - T_sol_120[index1]) * fraction1)/100);
    break;
default:
    interr1 = 0;
    interr2 = 0;
    break;
}
conc = (interr1 + ((interr2 - interr1) * fraction2)/100)/100;
}
return conc;
}

```