#include <stdio.h>

#include <math.h>

float A, B;

int sel\_;

float a1, b1a, b1b, c1;

float A11, A12, A13, A14, A15, A16, A17;

float M1, M2, M3, M4, M5, M6, M7;

float fucurah\_hujan[4] = {0, 0, 10, 100};

float fulama\_hujan[3] = {50, 150, 250};

float fudebit\_sungai[4] = {200, 290, 300, 300};

float fysiaga3[3] = {0, 0, 100};

float fysiaga2[3] = {20, 127.5, 235};

float fysiaga1[3] = {155, 255, 255};

float inputf, outputf;

float FiN(){

if (inputf < fucurah\_hujan[2]){return 1;}

else if (inputf >= fucurah\_hujan[2] && inputf <= fucurah\_hujan[3]){return (fucurah\_hujan[3] - inputf)/(fucurah\_hujan[3]-fucurah\_hujan[2]);}

else if (inputf > fucurah\_hujan[3]){return 0;}

}

float FiZ(){

if (inputf < fulama\_hujan[0]){return 0;}

else if (inputf >= fulama\_hujan[0] && inputf <= fulama\_hujan[1]){return (inputf - fulama\_hujan[0])/(fulama\_hujan[1]-fulama\_hujan[0]);}

else if (inputf >= fulama\_hujan[1] && inputf <= fulama\_hujan[2]){return (fulama\_hujan[2] - inputf)/(fulama\_hujan[2]-fulama\_hujan[1]);}

else if (inputf > fulama\_hujan[2]){return 0;}

}

float FiP(){

if (inputf < fudebit\_sungai[0]){return 0;}

else if (inputf >= fudebit\_sungai[0] && inputf <= fudebit\_sungai[1]){return (inputf - fudebit\_sungai[0])/(fudebit\_sungai[1]-fudebit\_sungai[0]);}

else if (inputf > fudebit\_sungai[1]){return 1;}

}

float FoN(){

if (outputf < fysiaga3[1]){return 1;}

else if (outputf >= fysiaga3[1] && outputf <= fysiaga3[2]){return (fysiaga3[2] - outputf)/(fysiaga3[2]-fysiaga3[1]);}

else if (outputf > fysiaga3[2]){return 0;}

}

float FoZ(){

if (outputf < fysiaga2[0]){return 0;}

else if (outputf >= fysiaga2[0] && outputf <= fysiaga2[1]){return (outputf - fysiaga2[0])/(fysiaga2[1]-fysiaga2[0]);}

else if (outputf >= fysiaga2[1] && outputf <= fysiaga2[2]){return (fysiaga2[2] - outputf)/(fysiaga2[2]-fysiaga2[1]);}

else if (outputf > fysiaga2[2]){return 0;}

}

float FoP(){

if (outputf < fysiaga1[0]){return 0;}

else if (outputf >= fysiaga1[0] && outputf <= fysiaga1[1]){return (outputf - fysiaga1[0])/(fysiaga1[1]-fysiaga1[0]);}

else if (outputf > fysiaga1[1]){return 1;}

}

void implikasi (){

a1 = 100 - (FiN() \* (fysiaga3[2] - fysiaga3[1]));

b1a = 20 + (FiZ() \* (fysiaga2[1] - fysiaga2[0]));

b1b = 235 - (FiZ() \* (fysiaga2[2] - fysiaga2[1]));

c1 = 155 + (FiP() \* (fysiaga1[1] - fysiaga1[0]));

}

void luas\_deffuzzy(){

implikasi ();

A11 = ((fysiaga3[2] - a1) \* FiN()) / 2;

A12 = (a1 - fysiaga3[0]) \* FiN();

A13 = ((b1a - fysiaga2[0])\* FiZ()) / 2;

A14 = ((fysiaga2[2] - b1b)\* FiZ()) / 2;

A15 = (b1b - b1a)\*FiZ();

A16 = ((c1 - fysiaga1[0]) \* FiP()) / 2;

A17 = (fysiaga1[2] - c1)\* FiP();

}

float f(float x){

if (B > 0 && sel\_ == 0){

return ((x-A)/B)\*x;

}

else if (B > 0 && sel\_ == 1){

return ((A-x)/B)\*x;

}

else {

return A\*x;

}

}

float simpsons(float f(float x), float a,float b,float n){

float h,integral,x,sum=0;

int i;

h=fabs(b-a)/n;

for(i=1;i<n;i++){

x=a+i\*h;

if(i%2==0){

sum=sum+2\*f(x);

}

else{

sum=sum+4\*f(x);

}

}

integral=(h/3)\*(f(a)+f(b)+sum);

return integral;

}

float fx(float limd, float limu, float a, float b, int sel){

int n,i=2;

float h,x,integral,eps=0.1,integral\_new;

A = a;

B = b;

sel\_ = sel;

integral\_new=simpsons(f,limd,limu,i);

do{

integral=integral\_new;

i=i+2;

integral\_new=simpsons(f,limd,limu,i);

}while(fabs(integral\_new-integral)>=eps);

return integral\_new;

}

void moment(){

luas\_deffuzzy();

M1 = fx(a1, fysiaga3[2], fysiaga3[2], (fysiaga3[2]-fysiaga3[0]), 1);

M2 = fx(fysiaga3[0], a1, FiN(), 0, 0);

M3 = fx(fysiaga2[0], b1a, fysiaga2[0], (fysiaga2[1] - fysiaga2[0]), 0);

M4 = fx (b1b, fysiaga2[2], fysiaga2[2], (fysiaga2[2]-fysiaga2[1]), 1);

M5 = fx (b1a, b1b, FiZ(), 0, 0);

M6 = fx(fysiaga1[0], c1, fysiaga1[0], (fysiaga1[2]-fysiaga1[0]), 0);

M7 = fx(c1, fysiaga1[2], FiP(), 0, 0);

}

float deffuzzyfikasi(){

return (M1+M2+M3+M4+M5+M6+M7)/(A11+A12+A13+A14+A15+A16+A17);

}

int main(){

printf ("Masukan Input Jarak 0 - 300 :");

scanf ("%f", &inputf);

printf ("Curah Hujan : %f\n", FiN());

printf ("Lama Hujan: %f\n", FiZ());

printf ("Debit Sungai: %f\n", FiP());

moment();

printf("a1 : %f\n", a1);

printf("b1a : %f\n", b1a);

printf("b1b : %f\n", b1b);

printf("c1 : %f\n", c1);

printf("A11 :%f\n", A11);

printf("A12 :%f\n", A12);

printf("A13 :%f\n", A13);

printf("A14 :%f\n", A14);

printf("A15 :%f\n", A15);

printf("A16 :%f\n", A16);

printf("A17 :%f\n", A17);

printf("M1 : %f\n", M1);

printf("M2 : %f\n", M2);

printf("M3 : %f\n", M3);

printf("M4 : %f\n", M4);

printf("M5 : %f\n", M5);

printf("M6 : %f\n", M6);

printf("M7 : %f\n", M7);

printf("OutDefuzzyfikasi : %f", deffuzzyfikasi());

}