1)For given sample test case

Program:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int decode\_value(int base, const char\* value) {

return strtol(value, NULL, base);

}

double lagrange\_interpolation(int x[], int y[], int num\_points) {

double result = 0;

for (int i = 0; i < num\_points; i++) {

double term = y[i];

for (int j = 0; j < num\_points; j++) {

if (j != i) {

term \*= (0.0 - x[j]) / (x[i] - x[j]);

}

}

result += term;

}

return result;

}

int main() {

int x[] = {1, 2, 3, 6};

const char\* y\_encoded[] = {"4", "111", "12", "213"};

int base[] = {10, 2, 10, 4};

int y[4];

int num\_points = sizeof(x) / sizeof(x[0]);

for (int i = 0; i < num\_points; i++) {

y[i] = decode\_value(base[i], y\_encoded[i]);

}

double c = lagrange\_interpolation(x, y, num\_points);

printf("The constant term (secret) c is: %lf\n", c);

return 0;

}

Output:The constant term (secret) c is: 3.000000

2)For test case2:

Program:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

long long decode\_value(int base, const char\* value) {

return strtoll(value, NULL, base);

}

double lagrange\_interpolation(int x[], long long y[], int num\_points) {

double result = 0;

for (int i = 0; i < num\_points; i++) {

double term = y[i];

for (int j = 0; j < num\_points; j++) {

if (j != i) {

term \*= (0.0 - x[j]) / (x[i] - x[j]);

}

}

result += term;

}

return result;

}

int main() {

int x[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};

const char\* y\_encoded[] = {

"28735619723837",

"1A228867F0CA",

"32811A4AA0B7B",

"917978721331A",

"1A22886782E1",

"28735619654702",

"71AB5070CC4B",

"122662581541670",

"642121030037605"

};

int base[] = {10, 16, 12, 11, 16, 10, 14, 9, 8};

long long y[9];

int num\_points = sizeof(x) / sizeof(x[0]);

for (int i = 0; i < num\_points; i++) {

y[i] = decode\_value(base[i], y\_encoded[i]);

}

double c = lagrange\_interpolation(x, y, num\_points);

printf("The constant term (secret) c is: %lf\n", c);

return 0;

}

Output:The constant term (secret) c is: 33198405829207.531250