

void change

N4

српавица 2

$$A \xrightarrow{+1} D \xrightarrow{+10} I \xrightarrow{+2} K$$

расстояние от A до K равно 12

Объем: 13; A, D, I, K.

N7

Объем: 1125.

N10

//C++

using namespace std;

vector<int> subField(8);

vector<vector<int>> field(8, subField);

int dist=100;

void add() { bool isFound=false;

for (int i=0; i<8; i++) {

for (int j=0; j<8; j++) {

if (i-1<0) {

if (j-1<0) {

if (j+1>7) {

if ((field[i][j]==0) && (field[i-1][j]==0) &&

(field[i][j-1]==0) && (field[i][j+1]==0)) {

isFound=true;

field[i][j]=8; field[i-1][j]=8; field[i][j-1]=8;

field[i][j+1]=8;

}

}

} if (!isFound) { cout << "Нем досягнуто метки" << endl; }

else { cout << "спингва віддалена" << endl; }

}

```

void show() {
    for (int i=0; i<8; i++) {
        for (int j=0; j<8; j++) {
            cout << field[i][j];
        }
        cout << endl;
    }
}

```

Число 3

```

void distance() {
    cout << "Начинаем движение с координат (0,0)";
    cin >> dist;
}

```

```

void move(int x, int y, string msg) {
    for (int i=0; i<8; i++) {
        for (int j=0; j<8; j++) {
            int newI = i+y;
            int newJ = j+x;
            if (newI > 8 || newI < 0 || newJ > 8 || newJ < 0) {
                break;
            }
            if (newI < 0 || newI > 8 || newJ < 0 || newJ > 8) {
                break;
            }
            if (newJ < 0 || newJ > 8 || newI < 0 || newI > 8) {
                break;
            }
            field[newI][newJ] = round(8 * (dist/100)); // round - округление
            field[newI][newJ] = min(field[newI][newJ], 8);
            field[i][j] = round(8 * (dist/100));
            field[i][j] = max(field[i][j], 0);
        }
    }
    cout << msg << endl;
}

```

```

void left() { move(-1, 0; "left"); }
void right() { move(1, 0; "right"); }
void up() { move(0, -1; "up"); }
void down() { move(0, 1; "down"); }

```

Empunyai 4

```

while (true)
{
    cin >> line;
    while (line)

```

```

    string line;
    while (true)
    {
        if (line == "add") { add(); }
        if (line == "show") { show(); }
        if (line == "up") { up(); }
        if (line == "down") { down(); }
        if (line == "left") { left(); }
        if (line == "right") { right(); }
        if (line == "distance") { distance(); }
    }
}

```

//nya benag ay yaka kura ke grapho
//b yk kudu

```

return 1;
}

```

N1 (3)

- 1) 3 dimensi ke 1 dimensi penerjemahan ke 2³=8>6)
 $X_p \Rightarrow$ penerjemahan penerjemahan.
 $X_{p1} = \frac{230 \cdot 470 \cdot 3}{8} = 405375 \text{ datum} \approx 40538 \text{ datum} - 1 \text{ cm penerjemahan penerjemahan}$
- 2) $35 \cdot 2^3 = X_p + \frac{X_p}{60} = \frac{61}{60} X_p = \frac{61 \cdot 47 \cdot X_{p1}}{60}$
 $Y_p \approx 63,6 \approx 63 - \text{cm penerjemahan penerjemahan}$
- 3) $X_T = \frac{X_p}{60} = 676 \text{ datum}$
 $Y_T = \frac{X_p - 8}{16 \cdot 40} = 47 - \text{cm penerjemahan penerjemahan}$
- 4) $Y_T + Y_p = 63 + 47 = 110 \text{ cm penerjemahan penerjemahan}$

Jawab: 110.

cin > 0 & 5 C kode karakter 0-127
 cout << "kode karakter 0-127: 0-127" << endl; // kode karakter 0-127
 << "kode karakter 0-127: 0-127" << endl; // kode karakter 0-127
 << "kode karakter 0-127: 0-127" << endl; // kode karakter 0-127

N9

Chp 5

using namespace std;

struct node { int temp; int r; int h; int m;

vector<node> nodes(128);

int nodeCounter = 0;

void init();

for (int i = 0; i < 128; i++) {

~~nodes[i] = node~~ n = new node();

n->r = 250;

}

}

void add();

void cycles();

cout << "Program telah selesai" << endl;

cin >> num;

cout << "Berikut" << num << " adalah representasi " << nodes[num].r << "px" << endl;

}

void add();

nodeCounter++;

if (nodeCounter > 2 * 128 * 250) { cout << "Terlalu banyak" << endl; return; }

cout << "Program telah berjalan 6 detik" << endl;

cin >> nodes[nodeCounter % 128].h;

cout << "Berapa banyak simpul maksimum" << endl;

cin >> nodes[nodeCounter % 128].m;

cout << "Berapa simpul" << nodes[nodeCounter % 128].h << " " <<

<< nodes[nodeCounter % 128].m << endl;

<< "Berapa simpul maksimum & jumlah simpul" << endl;

cin >> nodes[nodeCounter % 128].temp;

cout << "Berapa simpul maksimum: 6" << " " << nodes[nodeCounter % 128].temp << " " <<

<< nodes[nodeCounter % 128].m << " " << nodes[nodeCounter % 128].temp << endl;

}

