```
#include <iostream>
#include <cmath>
/*
Muvelet osztaly (class) dinamikus helyfoglalassal
*/
using namespace std;

class Muvelet
{
    public:
       float *x, *y; //adattagok, pointerek
       float *eredmeny;
       char *muvelet;
    public:
       //konstruktor - neve megegyezik a class nevevel, nincs
tipusa
       Muvelet(char muvelet1, float x1, float y1);
       ~Muvelet(); //destruktor!!!
       void Szamol();
       void Kiir();
};   //a pontosvesszô itt kötelezô

//implementáció következik, részletes leírása a
//konstruktörnek, destruktornak, Szamol()-nak, Kiir()-nak

//konstruktör:
Muvelet::Muvelet(char muvelet1, float x1, float y1)
{
    x = new float; //egy ujkent felvett valos tizedes valtozo
    y = new float;
    eredmeny = new float;
    muvelet = new char;
    *muvelet = muvelet1;
    *x = x1;
    *y = y1;
    *eredmeny = 99999999; //minden pointer erteket kap
}

//destruktor:
Muvelet::~Muvelet()
{
    delete x;
    delete y;
    delete eredmeny;
    delete muvelet;
}

void Muvelet::Szamol()
{
    switch (*muvelet)
    {
        case '+': *eredmeny = *x + *y; break;
        case '-': *eredmeny = *x - *y; break;
```

```
        case '*': *eredmeny = (*x) * (*y); break;
        case '/': if (*y!=0) *eredmeny = *x / *y; break;
        case '#': *eredmeny = (*x + *y)/2; break;
        case '^': *eredmeny = pow(*x, *y); break;
    }
}

void Muvelet::Kiir()
{
    cout<<"Eredmeny = "<<*eredmeny<<endl;
}

int main()
{
    cout << "Muvelet osztaly (class)" << endl;
    Muvelet *p; //a muvelet objektumra mutato pointer
    char muv;
    float X, Y;
    cout<<"X="; cin>>X;
    cout<<"Y="; cin>>Y;
    cout<<"Muvelet (+, -, *, /, #, ^): "; cin>>muv;
    p = new Muvelet(muv, X, Y);  //új "Muvelet" konstruálása
    p->Szamol();
    p->Kiir();
    delete p; //torlom a memoriabol a p-t objektumot
    return 0;
}
```