A házi nem más, mint egy aritmetika kód random generálása, méghozzá rekurzió segítségével. Ezt olyan bonyolult, mint amilyennek hangzik, scak annyit kell csinálni, hogy készítsetek egy random generátort (aminek beadtok egy magot, pl. srand()-al), és adjatok meg neki paraméterként valószínűségeket. Ezek a valószínűségek fogják eldönteni, hogy a soron következő karakter egy műveleti jel vagy bemenet/konstans lesz. (És pl. zárójelben lesz-e.) Az, hogy milyen műveleti jel legyen pl. az már egy egyszerű case-switch a random generátoron belül.

Pl:

void generator (int jel){

int value = rand()%10;

if(value <= jel){

\*Kezdjen új műveletet\*

}else{

\*Rakjon be konstanst vagy bemenetet\*

}

Ha a jel = 2, akkor például 30%-os eséllyel kapsz új műveletet, és 70%-al számot.

Attól függően, hogy hogyan paraméterezitek a függvényt, rövidebb-hosszabb kódokat fogtok kapni. Értelemszerűen akkor van vége a kódnak, ha minden műveletnek megvan a két operandusa.

Pár megjegyzés:

Viszonylag egyszerű olyan kódot írni, ami azonnal ki is számolja a megadott műveleteket (á la számológép program), és így könnyű lesz ellenőrizni a beadandótok kimeneteit.

Ami magát a generálást illeti, lehet, hogy a standard műveleti jelölés nem mindenkinek a legegyszerűbb, ezért mindenkinek ajánlom, hogy nézzen utána a prefix (lengyel)/infix/postfix jelölésrendszernek. (Legalábbis én biztosan beleőrülnék, ha standard „jobbról balra” módszerrel kéne generálnom a kódot. Meg lehet csinálni, de szerintem nem kényelmes.)

Ez önmagában semmilyen segítséget nem nyújt sem a házihoz, sem a beadandóhoz. Minden rendszernek vannak előnyei és hátrányai, de egyik sem jelentősen egyszerűbb, mint a többi. Az már személyes preferencia kérdése, hogy ki mivel dolgozik.

Még annyit tudok ajánlani, hogy nézzetek utána a műveleti fák építésének. Fában sokkal egyszerűbb a műveletek kiépítése, és a későbbi „olvasható formába” való kiírásuk is. Erre nincs feltétlenül szükség sem a házihoz, sem a beadandóhoz, de aki érzi magában az erőt, annak ez még jól jöhet.