# Դինամիկ զանգված

Զանգվածը կանվանենք *դինամիկ*, եթե այդ զանգվածի էլէեմենտները պահպանելու համար հիշողությունը հատկացվում է ծրագրի կատարման ընթացքում և կարող է հեռացվել:

*A = new int [n]; // ստեղծում ենք զանգված օբյեկտը, հատկացնում ենք հիշողություն*

*for (i = 0; i < n ; ++i)*

*{*

*A[i] = i \* i; // լրացնում ենք արժեքներով և արտածում*

*cout << A[i] << " ";*

*}*

*delete [] A; // ազատում ենք հիշողությունը*

*return 0;*

*}*

Մենք կարող ենք նույն կերպ ստեղծել դինամիկ փոփոխականներ և հեռացնել դրանք ծրագրի ընթացքում: Հիշողության դինամիկ հատկացումը երբեմն անհրաժեշտ է համակարգչի հիշողության արդյունավետ օգտագործմա համար: Օրինակ, եթե ծրագիրը մշակում է որևէ զանգված, և այն հայտարարվում է ֆիքսված չափով (օրինակ, մինչև 100 տարր), ապա այդ ծրագիրը չի կարող մշակել 100-ից ավել տարրով էլեմենտներից կազմված զանգված, իսկ 20 երկարության զանգվածի հետ աշխատելիս ավելի մեծ հիշողության հատկացումը լինում է անարդյունավետ:

C++-ում new և delete հրամանները հարավորություն են տալիս համակարգչի հիշողությունը դինամիկ բաշխել: new -ի օգնությամբ հիշողություն է հատկացվում համակարգչի ազատ հիշողությունից, իսկ delete -ը ազատում է այդ հիշողությունը դինամիկ փոփոխականի օգտագործումից հետո:

*int \*ptrvalue = new int; // ptrvalue ցուցիչը int տիպի փոփոխականին հատկացված*

*//հիշողությանն է ցույց տալիս*

*\*ptrvalue =10; // փոփոխականին արժեք է տրվում*

Արժեք կարելի է տալ նաև դինամիկ փոփոխական ստեղծելիս, օրինակ՝

*int \*ptrvalue = new int (10);*

*cout << \*ptrvalue << endl;*

*delete ptrvalue; // հիշողությունն ազատվում է:*

Սակայն սովորաբար new և delete հրամանները օգտագործվում են դինամիկ զանգված ստեղծելու համար:

*float \*ptrarray = new float [10];*

*…….*

*delete [ ] ptrarray;*

delete հրամանից հետո [ ] փակագծերը տեղեկացնում են, որ ազատվում է միաչափ զանգվածին հատկացված հիշողության տիրույթ: Հաջորդ օրինակում ստեղծվում է դինամիկ զանգված, լրացվում է պատահական իրական թվերով, հետո հեռացվում է:

// <ctime> վերնագրային ֆայլում պահվում է time() ֆունկցիայի նախատիպը

#include <ctime>

// <iomanip> վերնագրային ֆայլում պահվում է setprecision() ֆունկցիայի նախատիպը

#include <iomanip>

using namespace std;

int main(int argc, char\* argv[])

{

    srand(time(0)); // պատահական թվերի ստեղծում

    float \*ptrarray = new float [10]; // 10 իրական տարրերից կազմված դինամիկ զանգվածի ստեղծում

        for (int count = 0; count < 10; count++)

            ptrarray[count] = (rand() % 10 + 1) / float((rand() % 10 + 1)); //զանգվածը լրացվում է 1-ից 10 միջակայքի պատահական թվերով

        cout << "array = ";

        for (int count = 0; count < 10; count++)

            cout << setprecision(2) << ptrarray[count] << "    ";

        delete [] ptrarray; // հիշողությունն ազատվում է

        cout << endl;

    return 0;

}

## Երկչափ դինամիկ զանգվածի ստեղծման օրինակ

// 10 տարրից կազմված երկչափ դինամիկ զանգված

float \*\*ptrarray = new float\* [2]; // 2 տողից է կազմված

    for (int count = 0; count < 2; count++)

        ptrarray[count] = new float [5]; // 5 սյունակից

//  ptrarray  – float տիպի տարրերին հատկացված հիշողություներին դիմող ցուցիչների զանգված

// երկչափ զանգվածիհատկացվող հիշողության ազատում

    for (int count = 0; count < 2; count++)

        delete [] ptrarray[count];

//      2-ը զանգվածի տողերի քանակն է