#include<iostream>

using namespace std;

//

//void changeData(const int\* ptr) {

// cout << \*ptr << endl;

//}

//void main() {

// /\*int value = 100;

// cout << value << endl;

// const int\* ptr = &value;

// changeData(ptr);

// cout << value << endl;\*/

//

// /\*int arr[5] = { 1,2,3,4,5 };

// int\* ptr=arr;

// int\* ptr2=arr;

// ptr2 = NULL;

// for (size\_t i = 0; i < 5; i++)

// {

// cout << \*(ptr + i) << " ";

// }\*/

//

//

//

// /\*int arr[5] = { 1,2,3,4,5 };

// int\* ptr = arr;

// ++ptr;

// ptr += 3;

// ++(\*ptr);

// cout << \*ptr << endl;\*/

//

// //cout << \*(&\*(ptr + 3) - 2) << endl;

//

//

//

//

// /\*int value = 10000;

//

// int\* ptr = &value;

//

// \*ptr = 0;

// cout << value << endl;\*/

//

//

// //1.Pointer to const value

//

// //const int value = 10000;

// //const int\* ptr = &value;

//

// ////\*ptr = 0;//error

//

// //int value2 = 200;

// //ptr = &value2;

//

//

//

// //2.Const Pointer to value

//

// /\*int value = 10000;

// int value2 = 20000;

// int\*const ptr = &value;

// \*ptr = 5000;

// ptr = &value2;

// cout << \*ptr << endl;\*/

//

//

//

// //Const pointer to const value

//

// /\*const int value = 10000;

// const int\* const ptr = &value;

// int a = 10;\*/

// //\*ptr = 1000;//error

// //ptr = &a;//error

//

//

//

//

//

//

//

//}

//void show() {

// static int no = 10;

// ++no;

// cout << no << endl;

//}

//

//void main() {

// show();

// show();

// show();

//}

//

//void main() {

//

// /\*int\* ptr = new int(200);

//

// cout << \*ptr << endl;

// delete ptr;

// ptr = nullptr;

// cout << \*ptr << endl;\*/

//

//

// //double\* ptr = new double(10.2);

// //cout << \*ptr << endl;

// //ptr = NULL;//MEMORY LEAK

// /\*ptr = nullptr;

// cout << \*ptr << endl;\*/

//

//

//

//

// //int\* arr = new int[3]{1,2,3};

//

// //for (size\_t i = 0; i < 3; i++)

// //{

// // cout << arr[i] << " ";

// //}

//

//

// /\*int size = 0;

// cout << "enter size : ";

// cin >> size;

// int\* arr = new int[size]{};

// for (size\_t i = 0; i < size; i++)

// {

// arr[i] = i + 10;

// }

// for (size\_t i = 0; i < size; i++)

// {

// cout << arr[i] << " ";

// }

//

// delete[]arr;

// arr = nullptr;\*/

//

//

//}

int\* reserveArray(int \*size) {

int\* arr = new int[\*size] {};

return arr;

}

void initArray(int\* arr, int \*size) {

int min = 1;

int max = 100;

int random = 0;

for (size\_t i = 0; i < \*size; i++)

{

random = min + rand() % (max - min);

arr[i] = random;

}

}

void printArray(int\* arr, int \*size) {

for (size\_t i = 0; i < \*size; i++)

{

cout << arr[i] << " ";

}

cout << endl;

}

int\* pushBack(int\* arr, int \*size, int number) {

int\* newarray = new int[\*size + 1]{};

for (size\_t i = 0; i < \*size; i++)

{

newarray[i] = arr[i];

}

newarray[\*size] = number;

delete[]arr;

arr = newarray;

newarray = nullptr;

++(\*(size));

return arr;

}

int\* pushFront(int\* arr, int\* size, int number) {

int\* newarray = new int[\*size + 1]{};

newarray[0] = number;

for (size\_t i = 1; i < \*size+1; i++)

{

newarray[i] = arr[i - 1];

}

delete[]arr;

arr = newarray;

newarray = nullptr;

++(\*(size));

return arr;

}

void main() {

int size = 0;

cin >> size;

auto arr = reserveArray(&size);

initArray(arr, &size);

printArray(arr, &size);

arr=pushBack(arr, &size, 55);

arr=pushFront(arr, &size, 111);

printArray(arr, &size);

//gosterilen index e yeni elementi artiran funksiya yazin

}