
Vectors amb resize, i push i pop a front i back**X12746_ca**

En aquest exercici, heu d'implementar un programa que simula una estructura de dades que és una mena de `vector`, que admet accés indexat, però que també admet `push_front`, `push_back`, `pop_front`, `pop_back`. Inicialment, el vector se suposa buit. És possible modificar la seva mida amb els mètodes anteriors, però també amb el mètode `resize`, que preserva el valor de les posicions que continuen siguent vàlides, i fixa el valor de noves posicions a 0.

Més específicament, el vostre programa llegirà una sèrie de comandes d'entrada que s'executen sobre un vector *v* que se suposa inicialment buit. A continuació teniu un exemple d'entrada i a cada línia hi ha una explicació de com varia *v* i què s'escriu a la sortida:

```
v.push_back( 2 );    // v: 2
v.push_front( 1 );   // v: 1, 2
v.push_back( 3 );    // v: 1, 2, 3
v.resize( 5 );       // v: 1, 2, 3, 0, 0
cout<<v[ 0 ];        // output: 1
cout<<v[ 1 ];        // output: 2
cout<<v[ 3 ];        // output: 0
cout<<v;              // output: 1, 2, 3, 0, 0
v[ 4 ]= 6 ;          // v: 1, 2, 3, 0, 6
v.push_back( 5 );    // v: 1, 2, 3, 0, 6, 5
v.push_front( 4 );   // v: 4, 1, 2, 3, 0, 6, 5
cout<<v[ 0 ];        // output: 4
cout<<v[ 1 ];        // output: 1
cout<<v[ 3 ];        // output: 3
v.push_front( 8 );   // v: 8, 4, 1, 2, 3, 0, 6, 5
cout<<v[ 0 ];        // output: 8
cout<<v[ 1 ];        // output: 4
cout<<v[ 3 ];        // output: 2
cout<<v;              // output: 8, 4, 1, 2, 3, 0, 6, 5
v.resize( 3 );       // v: 8, 4, 1
v.resize( 6 );       // v: 8, 4, 1, 0, 0, 0
cout<<v[ 0 ];        // output: 8
cout<<v[ 1 ];        // output: 4
cout<<v[ 3 ];        // output: 0
cout<<v;              // output: 8, 4, 1, 0, 0, 0
v.pop_front();       // v: 4, 1, 0, 0, 0
v.pop_back();        // v: 4, 1, 0, 0
cout<<v[ 0 ];        // output: 4
cout<<v[ 1 ];        // output: 1
cout<<v[ 3 ];        // output: 0
cout<<v;              // output: 4, 1, 0, 0
```

Com veieu a l'exemple d'entrada anterior, hi han espais en blanc envoltant cada número per a facilitar la lectura de l'entrada. Per a facilitar-vos la lectura i el tractament de les comandes, us oferim una plantilla de programa que podeu completar:

```

#include <iostream>
#include <string>
// Add more includes if you wish.
// ...

using namespace std;

// Add auxiliary functions if you wish.
// ...

int main()
{
    // Declare and initialise more data if you wish.
    // ...
    string command;
    while (cin >> command) {
        if (command == "v.push_back(") {
            int val;
            cin >> val;
            string ending;
            cin >> ending; // Això consumeix el ";"
            // ...
        } else if (command == "v.push_front(") {
            int val;
            cin >> val;
            string ending;
            cin >> ending; // Això consumeix el ";"
            // ...
        } else if (command == "v.pop_front();") {
            // ...
        } else if (command == "v.pop_back();") {
            // ...
        } else if (command == "v.resize(") {
            int newsize;
            cin >> newsize;
            string ending;
            cin >> ending; // Això consumeix el ";"
            // ...
        } else if (command == "cout<<v[") {
            int index;
            cin >> index;
            string ending;
            cin >> ending; // Això consumeix el "];"
            // ...
        } else if (command == "cout<<v;") {
            // ...
        } else if (command == "v[") {
            int index;
            cin >> index;

```

```

        string mid;
        cin >> mid; // Això consumeix el "]"=
        int val;
        cin >> val;
        string ending;
        cin >> ending; // Això consumeix el ";"
        // ...
    } else if (command == "//") {
        string comment;
        getline(cin, comment);
        cout << "//" << comment << endl;
    } else {
        cout << "Wrong command '" << command << "'" << endl;
        string discard;
        getline(cin, discard);
    }
}
}

```

Us recomanem que comenceu implementant una solució senzilla que superi els jocs de proves públics, obtenint així la meitat de la nota, i que mireu d'optimitzar-la més tard, si teniu temps. Podeu utilitzar qualsevol de les estructures de dades presentades al curs (`vector`, `stack`, `queue`, `list`, `set`, `map`), i de la forma que considereu oportuna. Fixeu-vos, però, que enfocaments diferents donaran lloc a programes que seran més eficients o menys eficients, i d'això dependrà que pogueu superar només els jocs de proves públics o tots els jocs de proves, cosa que afectarà a la nota.

Entrada

L'entrada del programa és una seqüència de línies, a on cada línia conté una comanda que pot ser d'un dels següents tipus:

```

v.resize( NATURAL );
v.push_front( INTEGER );
v.push_back( INTEGER );
v.pop_front();
v.pop_back();
v[ INDEX ]= INTEGER ;
cout<<v[ INDEX ];
cout<<v;
// COMMENT

```

A on `INDEX` és un natural qualsevol entre 0 i el nombre d'elements que té el vector en aquell moment menys 1, `NATURAL` és un natural qualsevol, i `INTEGER` és un enter qualsevol.

Les comandes `// COMMENT` s'han de tornar a escriure tal qual per la sortida, i son comentaris que us poden ajudar a debugar el vostre codi.

Se suposa que les entrades son correctes. En particular, `pop_front` i `pop_back` només tindran lloc quan, en aquell moment, el vector tingui un o més elements.

Sortida

Per a cada instrucció `cout<<v[INDEX]`; el programa escriurà en una línia el que conté el vector a la posició indexada per `INDEX` en aquell moment. Per a cada instrucció `cout<<v`; el programa escriurà en una línia el contingut de `v` en aquell moment, separat per comes.

Exemple d'entrada 1

```
// v.push_back( 5 );
v.push_back( 5 );
// cout<<v;
cout<<v;
// v.push_back( 5 );
v.push_back( 5 );
// cout<<v;
cout<<v;
// cout<<v[ 1 ];
cout<<v[ 1 ];
// cout<<v;
cout<<v;
// v[ 0 ]= 7 ;
v[ 0 ]= 7 ;
// cout<<v;
cout<<v;
// v[ 1 ]= 3 ;
v[ 1 ]= 3 ;
// cout<<v;
cout<<v;
// v.push_front( 0 );
v.push_front( 0 );
// cout<<v;
cout<<v;
// cout<<v[ 2 ];
cout<<v[ 2 ];
// cout<<v;
cout<<v;
// v.push_front( 8 );
v.push_front( 8 );
// cout<<v;
cout<<v;
// v.push_back( 9 );
v.push_back( 9 );
// cout<<v;
cout<<v;
// v.push_back( 0 );
v.push_back( 0 );
// cout<<v;
cout<<v;
// v.push_back( 3 );
v.push_back( 3 );
// cout<<v;
cout<<v;
// v.push_front( 5 );
v.push_front( 5 );
// cout<<v;
cout<<v;
// v.resize( 2 );
v.resize( 2 );
// cout<<v;
cout<<v;
```

```
// v.push_back( 8 );
v.push_back( 8 );
// cout<<v;
cout<<v;
// cout<<v[ 0 ];
cout<<v[ 0 ];
// cout<<v;
cout<<v;
// v.push_back( 1 );
v.push_back( 1 );
// cout<<v;
cout<<v;
// cout<<v[ 0 ];
cout<<v[ 0 ];
// cout<<v;
cout<<v;
// v.push_front( 9 );
v.push_front( 9 );
// cout<<v;
cout<<v;
// v.push_back( 7 );
v.push_back( 7 );
// cout<<v;
cout<<v;
// v.pop_front();
v.pop_front();
// cout<<v;
cout<<v;
// v.pop_back();
v.pop_back();
// cout<<v;
cout<<v;
// v[ 2 ]= 3 ;
v[ 2 ]= 3 ;
// cout<<v;
cout<<v;
// v.resize( 1 );
v.resize( 1 );
// cout<<v;
cout<<v;
// v[ 0 ]= 3 ;
v[ 0 ]= 3 ;
// cout<<v;
cout<<v;
// cout<<v[ 0 ];
cout<<v[ 0 ];
// cout<<v;
cout<<v;
// v[ 0 ]= 5 ;
v[ 0 ]= 5 ;
// cout<<v;
cout<<v;
// v[ 0 ]= 6 ;
v[ 0 ]= 6 ;
```

```

// cout<<v;
cout<<v;
// cout<<v[ 0 ];
cout<<v[ 0 ];
// cout<<v;
cout<<v;
// v.push_front( 7 );
v.push_front( 7 );
// cout<<v;
cout<<v;
// v.push_front( 5 );
v.push_front( 5 );
// cout<<v;
cout<<v;
// v.pop_front();
v.pop_front();
// cout<<v;
cout<<v;
// v.push_front( 4 );
v.push_front( 4 );
// cout<<v;
cout<<v;
// v[ 2 ]= 4 ;
v[ 2 ]= 4 ;
// cout<<v;
cout<<v;
// cout<<v[ 1 ];
cout<<v[ 1 ];
// cout<<v;
cout<<v;
// v.push_front( 6 );
v.push_front( 6 );
// cout<<v;
cout<<v;
// cout<<v[ 3 ];
cout<<v[ 3 ];
// cout<<v;
cout<<v;
// v.push_front( 1 );
v.push_front( 1 );
// cout<<v;
cout<<v;
// cout<<v[ 2 ];
cout<<v[ 2 ];
// cout<<v;
cout<<v;
// cout<<v[ 1 ];
cout<<v[ 1 ];
// cout<<v;
cout<<v;
// v[ 2 ]= 6 ;
v[ 2 ]= 6 ;
// cout<<v;
cout<<v;
// v.push_front( 4 );
v.push_front( 4 );
// cout<<v;
cout<<v;
// cout<<v[ 4 ];
cout<<v[ 4 ];
// cout<<v;

```

```

cout<<v;
// v.push_back( 8 );
v.push_back( 8 );
// cout<<v;
cout<<v;
// v[ 5 ]= 7 ;
v[ 5 ]= 7 ;
// cout<<v;
cout<<v;
// v.push_front( 7 );
v.push_front( 7 );
// cout<<v;
cout<<v;
// v.push_back( 2 );
v.push_back( 2 );
// cout<<v;
cout<<v;
// v.push_back( 1 );
v.push_back( 1 );
// cout<<v;
cout<<v;
// cout<<v[ 5 ];
cout<<v[ 5 ];
// cout<<v;
cout<<v;
// v.push_front( 9 );
v.push_front( 9 );
// cout<<v;
cout<<v;
// v.push_back( 9 );
v.push_back( 9 );
// cout<<v;
cout<<v;
// v.push_front( 9 );
v.push_front( 9 );
// cout<<v;
cout<<v;
// cout<<v[ 4 ];
cout<<v[ 4 ];
// cout<<v;
cout<<v;
// v.pop_back();
v.pop_back();
// cout<<v;
cout<<v;
// v.pop_front();
v.pop_front();
// cout<<v;
cout<<v;
// v.push_back( 9 );
v.push_back( 9 );
// cout<<v;
cout<<v;
// cout<<v[ 3 ];
cout<<v[ 3 ];
// cout<<v;
cout<<v;
// v.push_front( 3 );
v.push_front( 3 );
// cout<<v;
cout<<v;

```

```

// v[ 12 ]= 6 ;
v[ 12 ]= 6 ;
// cout<<v;
cout<<v;
// v.pop_front();
v.pop_front();
// cout<<v;
cout<<v;
// v.push_front( 4 );
v.push_front( 4 );
// cout<<v;
cout<<v;
// v[ 5 ]= 2 ;
v[ 5 ]= 2 ;
// cout<<v;
cout<<v;
// v.push_back( 3 );
v.push_back( 3 );
// cout<<v;
cout<<v;
// cout<<v[ 10 ];
cout<<v[ 10 ];
// cout<<v;
cout<<v;
// v[ 3 ]= 0 ;
v[ 3 ]= 0 ;
// cout<<v;
cout<<v;
// v.push_front( 6 );
v.push_front( 6 );
// cout<<v;
cout<<v;
// v.push_front( 8 );
v.push_front( 8 );
// cout<<v;
cout<<v;
// v.push_back( 6 );
v.push_back( 6 );
// cout<<v;
cout<<v;
// v.pop_back();
v.pop_back();
// cout<<v;
cout<<v;
// v.pop_back();
v.pop_back();
// cout<<v;
cout<<v;
// cout<<v[ 0 ];
cout<<v[ 0 ];
// cout<<v;
cout<<v;
// v[ 8 ]= 1 ;
v[ 8 ]= 1 ;
// cout<<v;
cout<<v;
// v[ 13 ]= 0 ;
v[ 13 ]= 0 ;
// cout<<v;
cout<<v;
// v.push_back( 4 );

```

```

v.push_back( 4 );
// cout<<v;
cout<<v;
// v.push_front( 4 );
v.push_front( 4 );
// cout<<v;
cout<<v;
// cout<<v[ 16 ];
cout<<v[ 16 ];
// cout<<v;
cout<<v;
// v.push_front( 7 );
v.push_front( 7 );
// cout<<v;
cout<<v;
// v[ 9 ]= 6 ;
v[ 9 ]= 6 ;
// cout<<v;
cout<<v;
// v.push_back( 1 );
v.push_back( 1 );
// cout<<v;
cout<<v;
// cout<<v[ 4 ];
cout<<v[ 4 ];
// cout<<v;
cout<<v;
// cout<<v;
cout<<v;
// cout<<v;
cout<<v;

```

Exemple de sortida 1

```
// v.push_back( 5 );
// cout<<v;
5
// v.push_back( 5 );
// cout<<v;
5,5
// cout<<v[ 1 ];
5
// cout<<v;
5,5
// v[ 0 ]= 7 ;
// cout<<v;
7,5
// v[ 1 ]= 3 ;
// cout<<v;
7,3
// v.push_front( 0 );
// cout<<v;
0,7,3
// cout<<v[ 2 ];
3
// cout<<v;
0,7,3
// v.push_front( 8 );
// cout<<v;
8,0,7,3
// v.push_back( 9 );
// cout<<v;
8,0,7,3,9
// v.push_back( 0 );
// cout<<v;
8,0,7,3,9,0
// v.push_back( 3 );
// cout<<v;
8,0,7,3,9,0,3
// v.push_front( 5 );
// cout<<v;
5,8,0,7,3,9,0,3
// v.resize( 2 );
// cout<<v;
5,8
// v.push_back( 8 );
// cout<<v;
5,8,8
// cout<<v[ 0 ];
5
// cout<<v;
5,8,8
// v.push_back( 1 );
// cout<<v;
5,8,8,1
// cout<<v[ 0 ];
5
// cout<<v;
5,8,8,1
// v.push_front( 9 );
// cout<<v;
9,5,8,8,1
// v.push_back( 7 );
```

```
// cout<<v;
9,5,8,8,1,7
// v.pop_front();
// cout<<v;
5,8,8,1,7
// v.pop_back();
// cout<<v;
5,8,8,1
// v[ 2 ]= 3 ;
// cout<<v;
5,8,3,1
// v.resize( 1 );
// cout<<v;
5
// v[ 0 ]= 3 ;
// cout<<v;
3
// cout<<v[ 0 ];
3
// cout<<v;
3
// v[ 0 ]= 5 ;
// cout<<v;
5
// v[ 0 ]= 6 ;
// cout<<v;
6
// cout<<v[ 0 ];
6
// cout<<v;
6
// v.push_front( 7 );
// cout<<v;
7,6
// v.push_front( 5 );
// cout<<v;
5,7,6
// v.pop_front();
// cout<<v;
7,6
// v.push_front( 4 );
// cout<<v;
4,7,6
// v[ 2 ]= 4 ;
// cout<<v;
4,7,4
// cout<<v[ 1 ];
7
// cout<<v;
4,7,4
// v.push_front( 6 );
// cout<<v;
6,4,7,4
// cout<<v[ 3 ];
4
// cout<<v;
6,4,7,4
// v.push_front( 1 );
// cout<<v;
1,6,4,7,4
// cout<<v[ 2 ];
```

```

4
// cout<<v;
1,6,4,7,4
// cout<<v[ 1 ];
6
// cout<<v;
1,6,4,7,4
// v[ 2 ]= 6 ;
// cout<<v;
1,6,6,7,4
// v.push_front( 4 );
// cout<<v;
4,1,6,6,7,4
// cout<<v[ 4 ];
7
// cout<<v;
4,1,6,6,7,4
// v.push_back( 8 );
// cout<<v;
4,1,6,6,7,4,8
// v[ 5 ]= 7 ;
// cout<<v;
4,1,6,6,7,7,8
// v.push_front( 7 );
// cout<<v;
7,4,1,6,6,7,7,8
// v.push_back( 2 );
// cout<<v;
7,4,1,6,6,7,7,8,2
// v.push_back( 1 );
// cout<<v;
7,4,1,6,6,7,7,8,2,1
// cout<<v[ 5 ];
7
// cout<<v;
7,4,1,6,6,7,7,8,2,1
// v.push_front( 9 );
// cout<<v;
9,7,4,1,6,6,7,7,8,2,1
// v.push_back( 9 );
// cout<<v;
9,7,4,1,6,6,7,7,8,2,1,9
// v.push_front( 9 );
// cout<<v;
9,9,7,4,1,6,6,7,7,8,2,1,9
// cout<<v[ 4 ];
1
// cout<<v;
9,9,7,4,1,6,6,7,7,8,2,1,9
// v.pop_back();
// cout<<v;
9,9,7,4,1,6,6,7,7,8,2,1
// v.pop_front();
// cout<<v;
9,7,4,1,6,6,7,7,8,2,1
// v.push_back( 9 );
// cout<<v;
9,7,4,1,6,6,7,7,8,2,1,9
// cout<<v[ 3 ];
1
// cout<<v;

```

```

9,7,4,1,6,6,7,7,8,2,1,9
// v.push_front( 3 );
// cout<<v;
3,9,7,4,1,6,6,7,7,8,2,1,9
// v[ 12 ]= 6 ;
// cout<<v;
3,9,7,4,1,6,6,7,7,8,2,1,6
// v.pop_front();
// cout<<v;
9,7,4,1,6,6,7,7,8,2,1,6
// v.push_front( 4 );
// cout<<v;
4,9,7,4,1,6,6,7,7,8,2,1,6
// v[ 5 ]= 2 ;
// cout<<v;
4,9,7,4,1,2,6,7,7,8,2,1,6
// v.push_back( 3 );
// cout<<v;
4,9,7,4,1,2,6,7,7,8,2,1,6,3
// cout<<v[ 10 ];
2
// cout<<v;
4,9,7,4,1,2,6,7,7,8,2,1,6,3
// v[ 3 ]= 0 ;
// cout<<v;
4,9,7,0,1,2,6,7,7,8,2,1,6,3
// v.push_front( 6 );
// cout<<v;
6,4,9,7,0,1,2,6,7,7,8,2,1,6,3
// v.push_front( 8 );
// cout<<v;
8,6,4,9,7,0,1,2,6,7,7,8,2,1,6,3
// v.push_back( 6 );
// cout<<v;
8,6,4,9,7,0,1,2,6,7,7,8,2,1,6,3,6
// v.pop_back();
// cout<<v;
8,6,4,9,7,0,1,2,6,7,7,8,2,1,6,3
// v.pop_back();
// cout<<v;
8,6,4,9,7,0,1,2,6,7,7,8,2,1,6
// cout<<v[ 0 ];
8
// cout<<v;
8,6,4,9,7,0,1,2,6,7,7,8,2,1,6
// v[ 8 ]= 1 ;
// cout<<v;
8,6,4,9,7,0,1,2,1,7,7,8,2,1,6
// v[ 13 ]= 0 ;
// cout<<v;
8,6,4,9,7,0,1,2,1,7,7,8,2,0,6
// v.push_back( 4 );
// cout<<v;
8,6,4,9,7,0,1,2,1,7,7,8,2,0,6,4
// v.push_front( 4 );
// cout<<v;
4,8,6,4,9,7,0,1,2,1,7,7,8,2,0,6,4
// cout<<v[ 16 ];
4
// cout<<v;
4,8,6,4,9,7,0,1,2,1,7,7,8,2,0,6,4

```



```
// v.push_front( 7 );
// cout<<v;
7, 4, 8, 6, 4, 9, 7, 0, 1, 2, 1, 7, 7, 8, 2, 0, 6, 4
// v[ 9 ]= 6 ;
// cout<<v;
7, 4, 8, 6, 4, 9, 7, 0, 1, 6, 1, 7, 7, 8, 2, 0, 6, 4
// v.push_back( 1 );
// cout<<v;
7, 4, 8, 6, 4, 9, 7, 0, 1, 6, 1, 7, 7, 8, 2, 0, 6, 4, 1
```

Exemple d'entrada 2

```
// v.push_back( 3 );
v.push_back( 3 );
// cout<<v;
cout<<v;
// v.resize( 6 );
v.resize( 6 );
// cout<<v;
cout<<v;
// v.push_back( 9 );
v.push_back( 9 );
// cout<<v;
cout<<v;
// v.push_back( 2 );
v.push_back( 2 );
// cout<<v;
cout<<v;
// v.push_back( 0 );
v.push_back( 0 );
// cout<<v;
cout<<v;
// v.push_front( 3 );
v.push_front( 3 );
// cout<<v;
cout<<v;
// cout<<v[ 6 ];
cout<<v[ 6 ];
// cout<<v;
cout<<v;
// v.push_back( 6 );
v.push_back( 6 );
// cout<<v;
cout<<v;
// v.push_front( 8 );
v.push_front( 8 );
// cout<<v;
cout<<v;
// cout<<v[ 9 ];
cout<<v[ 9 ];
// cout<<v;
cout<<v;
// v.push_front( 2 );
v.push_front( 2 );
// cout<<v;
cout<<v;
// v[ 10 ]= 5 ;
v[ 10 ]= 5 ;
// cout<<v;
cout<<v;
// v.pop_back();
```

```
// cout<<v[ 4 ];
4
// cout<<v;
7, 4, 8, 6, 4, 9, 7, 0, 1, 6, 1, 7, 7, 8, 2, 0, 6, 4, 1
// cout<<v;
7, 4, 8, 6, 4, 9, 7, 0, 1, 6, 1, 7, 7, 8, 2, 0, 6, 4, 1
// cout<<v;
7, 4, 8, 6, 4, 9, 7, 0, 1, 6, 1, 7, 7, 8, 2, 0, 6, 4, 1
```

```
v.pop_back();
// cout<<v;
cout<<v;
// v.push_front( 2 );
v.push_front( 2 );
// cout<<v;
cout<<v;
// v.push_back( 9 );
v.push_back( 9 );
// cout<<v;
cout<<v;
// v.push_front( 3 );
v.push_front( 3 );
// cout<<v;
cout<<v;
// v.push_front( 1 );
v.push_front( 1 );
// cout<<v;
cout<<v;
// v.push_front( 9 );
v.push_front( 9 );
// cout<<v;
cout<<v;
// v.push_front( 1 );
v.push_front( 1 );
// cout<<v;
cout<<v;
// v.push_front( 4 );
v.push_front( 4 );
// cout<<v;
cout<<v;
// v.pop_back();
v.pop_back();
// cout<<v;
cout<<v;
// v.push_front( 4 );
v.push_front( 4 );
// cout<<v;
cout<<v;
// v.push_back( 0 );
v.push_back( 0 );
// cout<<v;
cout<<v;
// v.push_back( 6 );
v.push_back( 6 );
// cout<<v;
cout<<v;
// v.push_front( 0 );
v.push_front( 0 );
// cout<<v;
```

```

cout<<v;
// v.push_back( 3 );
v.push_back( 3 );
// cout<<v;
cout<<v;
// v.push_back( 0 );
v.push_back( 0 );
// cout<<v;
cout<<v;
// v.push_back( 1 );
v.push_back( 1 );
// cout<<v;
cout<<v;
// v[ 0 ]= 4 ;
v[ 0 ]= 4 ;
// cout<<v;
cout<<v;
// cout<<v[ 8 ];
cout<<v[ 8 ];
// cout<<v;
cout<<v;
// v.push_front( 9 );
v.push_front( 9 );
// cout<<v;
cout<<v;
// v[ 13 ]= 4 ;
v[ 13 ]= 4 ;
// cout<<v;
cout<<v;
// cout<<v[ 22 ];
cout<<v[ 22 ];
// cout<<v;
cout<<v;
// v.push_front( 7 );
v.push_front( 7 );
// cout<<v;
cout<<v;
// v.pop_front();
v.pop_front();
// cout<<v;
cout<<v;
// v.push_front( 3 );
v.push_front( 3 );
// cout<<v;
cout<<v;
// v.push_front( 7 );
v.push_front( 7 );
// cout<<v;
cout<<v;
// v.push_front( 6 );
v.push_front( 6 );
// cout<<v;
cout<<v;
// v.pop_front();
v.pop_front();
// cout<<v;
cout<<v;
// cout<<v;
cout<<v;
// cout<<v;
cout<<v;

```

Exemple de sortida 2

```

// v.push_back( 3 );
// cout<<v;
3
// v.resize( 6 );
// cout<<v;
3,0,0,0,0,0
// v.push_back( 9 );
// cout<<v;
3,0,0,0,0,0,9
// v.push_back( 2 );
// cout<<v;
3,0,0,0,0,0,9,2
// v.push_back( 0 );
// cout<<v;
3,0,0,0,0,0,9,2,0
// v.push_front( 3 );
// cout<<v;
3,3,0,0,0,0,9,2,0
// cout<<v[ 6 ];
0
// cout<<v;
3,3,0,0,0,0,9,2,0
// v.push_back( 6 );
// cout<<v;
3,3,0,0,0,0,9,2,0,6
// v.push_front( 8 );
// cout<<v;
8,3,3,0,0,0,9,2,0,6
// cout<<v[ 9 ];
2
// cout<<v;
8,3,3,0,0,0,9,2,0,6
// v.push_front( 2 );
// cout<<v;
2,8,3,3,0,0,9,2,0,6
// v[ 10 ]= 5 ;
// cout<<v;
2,8,3,3,0,0,9,5,0,6
// v.pop_back();
// cout<<v;
2,8,3,3,0,0,9,5,0
// v.push_front( 2 );
// cout<<v;
2,2,8,3,3,0,9,5,0
// v.push_back( 9 );
// cout<<v;
2,2,8,3,3,0,9,5,0,9
// v.push_front( 3 );
// cout<<v;
3,2,2,8,3,3,9,5,0,9
// v.push_front( 1 );
// cout<<v;
1,3,2,2,8,3,9,5,0,9
// v.push_front( 9 );
// cout<<v;
9,1,3,2,2,8,3,9,5,0,9
// v.push_front( 1 );
// cout<<v;
1,9,1,3,2,2,8,3,9,5,0,9

```

| | |
|--|--|
| <pre>// v.push_front(4); // cout<<v; 4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0,9 // v.pop_back(); // cout<<v; 4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0 // v.push_front(4); // cout<<v; 4,4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0 // v.push_back(0); // cout<<v; 4,4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0,0 // v.push_back(6); // cout<<v; 4,4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0,0,6 // v.push_front(0); // cout<<v; 0,4,4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0,0,6 // v.push_back(3); // cout<<v; 0,4,4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0,0,6,3 // v.push_back(0); // cout<<v; 0,4,4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0,0,6,3,0 // v.push_back(1); // cout<<v; 0,4,4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0,0,6,3,0,1 // v[0]= 4 ; // cout<<v; 4,4,4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0,0,6,3,0,1 // cout<<v[8]; 2 // cout<<v;</pre> | <pre>4,4,4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0,0,6,3,0,1 // v.push_front(9); // cout<<v; 9,4,4,4,1,9,1,3,2,2,8,3,3,0,0,0,0,0,9,5,0,0,6,3,0,1 // v[13]= 4 ; // cout<<v; 9,4,4,4,1,9,1,3,2,2,8,3,3,4,0,0,0,0,9,5,0,0,6,3,0,1 // cout<<v[22]; 6 // cout<<v; 9,4,4,4,1,9,1,3,2,2,8,3,3,4,0,0,0,0,9,5,0,0,6,3,0,1 // v.push_front(7); // cout<<v; 7,9,4,4,4,1,9,1,3,2,2,8,3,3,4,0,0,0,0,9,5,0,0,6,3,0,1 // v.pop_front(); // cout<<v; 9,4,4,4,1,9,1,3,2,2,8,3,3,4,0,0,0,0,9,5,0,0,6,3,0,1 // v.push_front(3); // cout<<v; 3,9,4,4,4,1,9,1,3,2,2,8,3,3,4,0,0,0,0,9,5,0,0,6,3,0,1 // v.push_front(7); // cout<<v; 7,3,9,4,4,4,1,9,1,3,2,2,8,3,3,4,0,0,0,0,9,5,0,0,6,3,0,1 // v.push_front(6); // cout<<v; 6,7,3,9,4,4,4,1,9,1,3,2,2,8,3,3,4,0,0,0,0,9,5,0,0,6,3,0,1 // v.pop_front(); // cout<<v; 7,3,9,4,4,4,1,9,1,3,2,2,8,3,3,4,0,0,0,0,9,5,0,0,6,3,0,1 // cout<<v; 7,3,9,4,4,4,1,9,1,3,2,2,8,3,3,4,0,0,0,0,9,5,0,0,6,3,0,1 // cout<<v; 7,3,9,4,4,4,1,9,1,3,2,2,8,3,3,4,0,0,0,0,9,5,0,0,6,3,0,1</pre> |
|--|--|

Observació

Les entrades dels jocs de proves grans tindran poques comandes del tipus `cout<<v;`, i apareixeran quan el vector sigui prou petit, de manera que aquesta mena de comandes no hauria d'afectar al temps d'execució, sempre i quan tinguin cost $O(m \log(m))$, a on m seria el nombre d'elements al vector en aquell moment.

Avaluació sobre 10 punts:

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta, de cost $O(n \log(n))$ (excloent el temps consumit per les comandes de tipus `cout<<v;`, i a on n és el nombre total de la resta de comandes) i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Atenció: De cara a superar tots els jocs de proves, no és imprescindible que el tractament de cada comanda diferent de `cout<<v;` de forma individualitzada tingui cost logarítmic (tot i que és possible aconseguir-ho), però sí cal que el cost global sigui $n \log(n)$.

Informació del problema

Autoria: PRO2

Generació: 2026-01-25T13:38:10.193Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>