

19. Хипуй

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В этой задаче вам необходимо самостоятельно (не используя соответствующие классы и функции стандартной библиотеки) организовать структуру данных `Heap` для хранения целых чисел, над которой определены следующие операции: а) `Insert(k)` – добавить в `Heap` число `k` ; б) `Extract` достать из `Heap` наибольшее число (удалив его при этом).

Формат ввода

В первой строке содержится количество команд `N` ($1 \leq N \leq 100000$), далее следуют `N` команд, каждая в своей строке. Команда может иметь формат: “0 <число>” или “1”, обозначающий, соответственно, операции `Insert(<число>)` и `Extract`. Гарантируется, что при выполнении команды `Extract` в структуре находится по крайней мере один элемент.

Формат вывода

Для каждой команды извлечения необходимо отдельной строкой вывести число, полученное при выполнении команды `Extract`.

Пример 1

Ввод <input type="text"/>	Вывод <input type="text"/>
2	10000
0 10000	
1	

Пример 2

Ввод <input type="text"/>	Вывод <input type="text"/>
14	345
0 1	4346
0 345	2435
1	365
0 4346	235
1	5
0 2435	1
1	
0 235	
0 5	
0 365	
1	
1	
1	
1	

Язык

GNU GCC 12.2 C++20

Набрать здесь

Отправить файл

```
1 #include <iostream>
2 #include <vector>
3
4 void insert(std::vector<int>& v) {
5     int idx = v.size() - 1;
6     int val;
7     while (true) {
8         if (idx < 0) break;
9         val = v[idx];
10        v[idx] = v[idx + 1];
11        v[idx + 1] = val;
12    }
13 }
```

```
6     while (idx > 0) {
7         if (v[idx] > v[(idx - 1) / 2]) {
8             std::swap(v[idx], v[(idx - 1) / 2]);
9             --idx;
10            idx /= 2;
11        } else {
12            break;
13        }
14    }
15 }
16
17 void extract(std::vector<int>& v) {
18     std::cout << v[0] << "\n";
19     v[0] = v[v.size() - 1];
20     int idx = 0;
21     while (idx < v.size()) {
22         if (idx * 2 + 2 < v.size()) {
23             if (v[idx] >= std::max(v[idx * 2 + 1], v[idx * 2 + 2])) {
24                 break;
25             }
26             if (v[idx * 2 + 1] < v[idx * 2 + 2]) {
27                 std::swap(v[idx * 2 + 2], v[idx]);
28                 idx = idx * 2 + 2;
29             } else {
30                 std::swap(v[idx * 2 + 1], v[idx]);
31                 idx = idx * 2 + 1;
32             }
33         } else {
34             break;
35         }
36     }
37     v.pop_back();
38 }
```

ОтправитьПредыдущаяСледующая