Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий и анализа данных

Отчет  
о лабораторной работе № 7

«Классические алгоритмы»

по дисциплине

**«Программирование»**

|  |  |
| --- | --- |
| Выполнил: студент(-ка) гр. | Распутин Г.М. |
| Руководитель: | Лебедева Е.Н. |
| Оценка: |  |
| Дата защиты: |  |

**Иркутск 2024**

### Задание 1.1 - 2^n

1. Условие задачи.

Напишите программу, вычисляющую заданную степень числа 2, используя битовые операции.

**Входные данные**

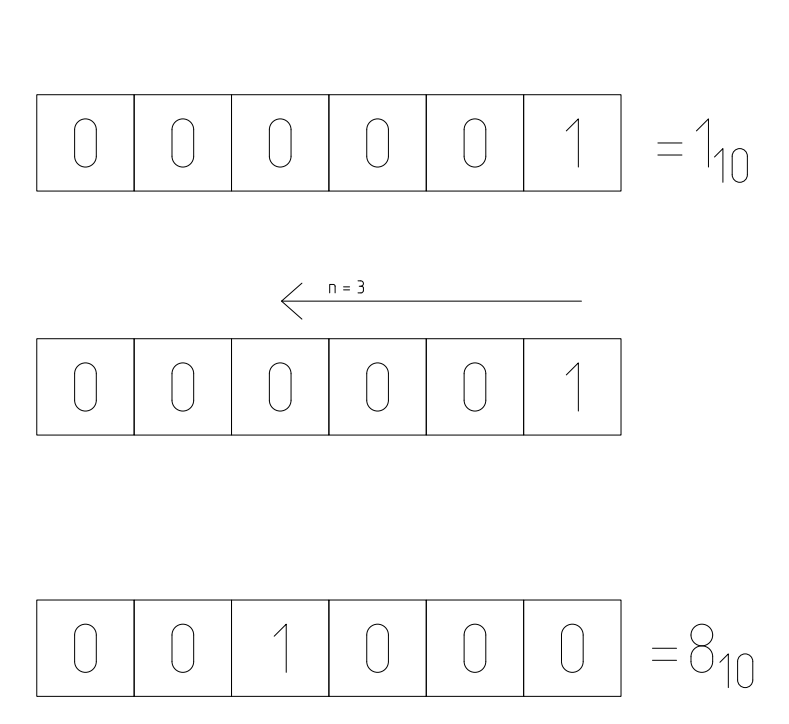
Дано число n<32.

**Выходные данные**

Выведите число 2n, то есть число, у которого n-й бит равен 1, а остальные – нули.

1. Пояснения решения

Для решения берём long long int переменную, чтобы записать туда единицу. Затем двигаем единицу на n разрядов влево.



1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить при малом N | N=5 | 32 |
| 2 | Проверить при большом N | N= 31 | 2147483648 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <iostream>

int main() {

int n;

std::cin >> n;

long long int oneee = 1;

long long int t = oneee<<n;

std::cout << t << "\n";

return 0;

}

### Задание 1.2 - 2^n+2^m

1. Условие задачи.

Найдите сумму двух различных степеней числа 2, используя только битовые операции. В частности нельзя использовать операцию сложения чисел.

**Входные данные**

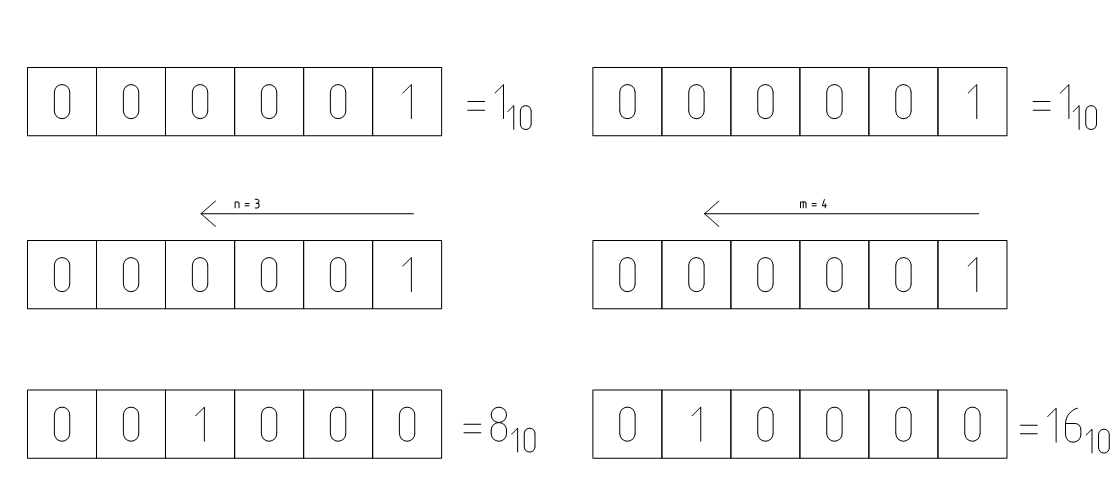
Даны два неравных числа: n и m, не превосходящие 31.

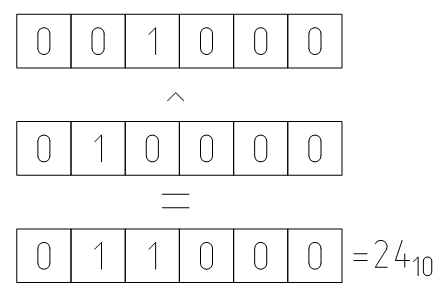
**Выходные данные**

Выведите на экран значение суммы 2n+2m.

1. Пояснения решения

Для решения берём две long long int переменные, чтобы записать туда единицы. Затем двигаем единицы на n и m разрядов влево. А затем складываем их с помощью операции поразрядное исключающее ИЛИ(^).





1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить при малом N и M | N=5  M = 3 | 40 |
| 2 | Проверить при большом N и M | N= 31  M = 30 | 3221225472 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <iostream>

int main() {

long long int n, m;

std::cin >> n >> m;

long long int oneee = 1;

long long int result = (oneee << n) ^ (oneee << m);

std::cout << result << "\n";

return 0;

}

### Задание 2.1 - Установить значение бита в 1

1. Условие задачи.

Напишите программу, которая в заданном числе устанавливает определенный бит в 1 (биты при этом нумеруются с нуля, начиная от младших).

**Входные данные**

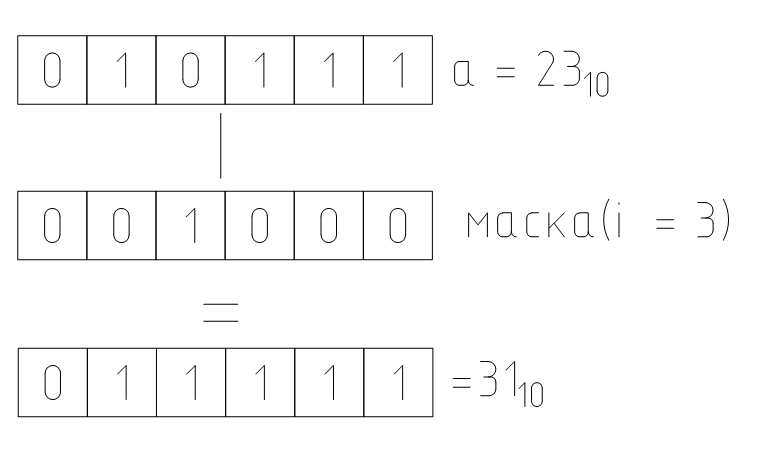
Дано целое число A и натуральное число i.

**Выходные данные**

Выведите число, которое получается из числа A установкой значения i-го бита = 1..

1. Пояснения решения

Для решения берём long long int переменную, чтобы записать туда единицу. Затем двигаем единицу на i разрядов влево - это будет маской. Используем эту маску чтобы установить в заданном числе определенный бит в один с помощью операции поразрядная дизъюнкция(|)



1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить при малом a и i | A=5  i = 3 | 13 |
| 2 | Проверить при большом a и i | A= 31  i = 30 | 1073741855 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <iostream>

int main() {

long long int oneee = 1;

long long int a, i;

std::cin >> a >> i;

long long int mask = oneee << i;

long long int res = a | mask;

std::cout << res << "\n";

return 0;

}

### Задание 2.2 - Установить значение бита в 0

1. Условие задачи.

Напишите программу, устанавливающую значение определенного бита числа в 0 (биты при этом нумеруются с нуля, начиная от младших).

**Входные данные**

Дано целое число A и натуральное число i.

**Выходные данные**

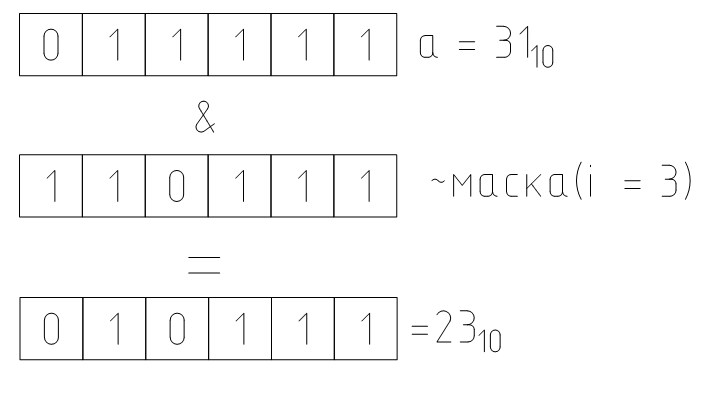
Выведите число, которое получается из числа A установкой значения i-го бита = 0.

1. Пояснения решения

Для решения берём long long int переменную, чтобы записать туда единицу. Затем двигаем единицу на i разрядов влево - это будет маской.

В отличии от прошлого решения – инвертируем маску с помощью операции поразрядное отрицание (~)

Используем эту маску чтобы установить в заданном числе определенный бит в один с помощью операции поразрядная конъюнкция (&)



1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить при малом a и i | A=3831  i = 4 | 3815 |
| 2 | Проверить при большом a и i | A= 16127  i = 7 | 15999 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <iostream>

int main() {

long long int oneee = 1;

long long int a, i;

std::cin >> a >> i;

long long int mask = oneee << i;

mask = ~mask;

long long int res = a & mask;

std::cout << res << "\n";

return 0;

}

### Задание 3 - Определить значение бита

1. Условие задачи.

Напишите программу, определяющую значение i-го бита числа (биты при этом нумеруются с нуля, начиная от младших).

**Входные данные**

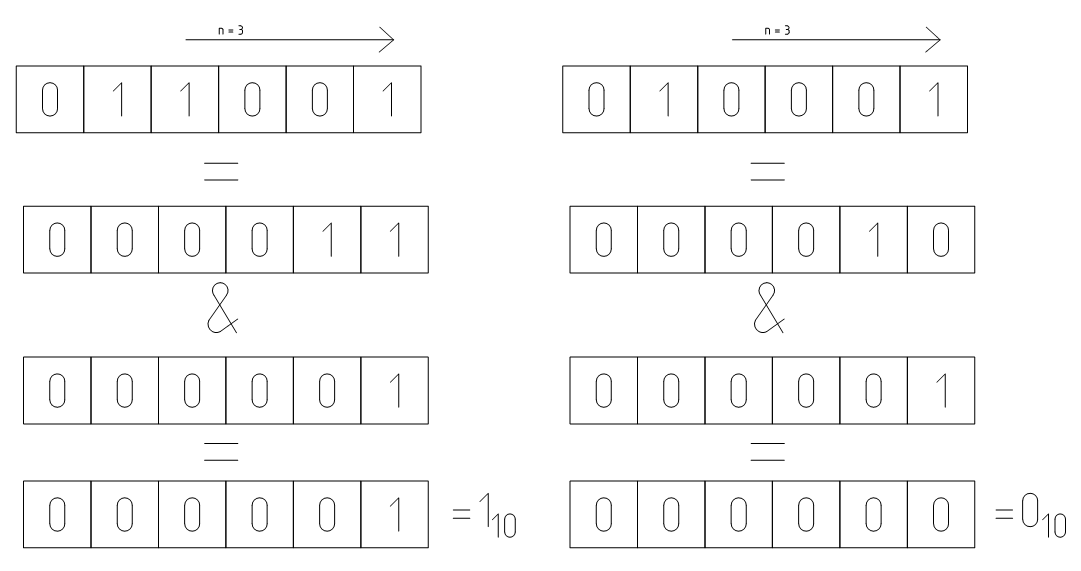
Дано целое число A и целое неотрицательное число i.

**Выходные данные**

Выведите значение i-го бита числа A, то есть 0 или 1.

1. Пояснения решения

Заданное число A двигаем на i разрядов вправо, а затем умножаем на 1 с помощью операции поразрядная конъюнкция (&)



1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить при малом a и i | A=31  i = 3 | 1 |
| 2 | Проверить при большом a и i | A= 16127  i = 7 | 1 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <iostream>

int main() {

long long int a, i;

std::cin >> a >> i;

long long int oneee = 1;

long long int result = (a >> i) & oneee;

std::cout << result << "\n";

return 0;

}

### Задание 4 - Алгоритм Евклида

1. Условие задачи.

По данным натуральным числам n и m найдите их наибольший общий делитель.

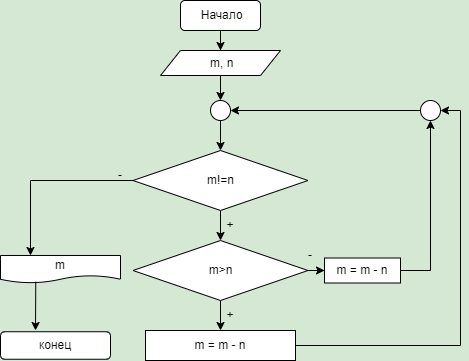
**Входные данные**

Программа получает на вход 2 натуральных числа m и n. Числа m и n не превосходят 109.

**Выходные данные**

Программа должна вывести наибольший общий делитель двух данных чисел.

1. Пояснения решения



1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить при малом m и n | M = 20  N = 30 | 10 |
| 2 | Проверить при большом m и n | M= 1436127  N = 12342 | 33 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <iostream>

using namespace std;

int main() {

int m, n;

cin >> m >> n;

while (m != n) {

if (m>n) {

m = m-n;

}

else {

n = n-m;

}

}

cout << m;

}

### Задание 4 - Из шестнадцатеричной в двоичную

1. Условие задачи.

Напишите программу, переводящую число из шестнадцатеричной системы счисления в двоичную.

**Входные данные**

Программа получает на вход строку, состоящую из цифр 0, ..., 9 и букв A, ..., F, являющуюся записью некоторого 16-ричного целого числа. Длина строки не превосходит 50 символов, первый символ в строке не равен 0. Необходимо вывести запись этого числа в двоичном виде без лидирующих нулей.

**Выходные данные**

Выведите результат перевода.

1. Пояснения решения

Каждому шестнадцатеричному числу соответствует четыре числа в двоичной системе. Можно составить словарь, ключами которого будут шестнадцатеричные символы, а значением – аналог на двоичной системе. Далее, просто пройти строку шестнадцатеричного числа, записывая в двоичной системе в новой переменной. Затем удалить ведущие нули.

1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить работоспособность | A123F123B | 101000010010001111110001001000111011 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <iostream>

#include <map>

using namespace std;

int main() {

string hstring;

cin >> hstring;

map <char, string> binhex = {

{'0', "0000"}, {'1', "0001"}, {'2', "0010"}, {'3', "0011"},

{'4', "0100"}, {'5', "0101"}, {'6', "0110"}, {'7', "0111"},

{'8', "1000"}, {'9', "1001"}, {'A', "1010"}, {'B', "1011"},

{'C', "1100"}, {'D', "1101"}, {'E', "1110"}, {'F', "1111"}

};

string bstring = "";

for (char n : hstring){

bstring += binhex[n];

}

while (bstring[0] == '0' && bstring.length() > 1) {

bstring = bstring.substr(1);

}

cout << bstring << "\n";

return 0;

}

### Задание 5 - Из двоичной в шестнадцатеричную

1. Условие задачи.

Напишите программу, переводящую число из двоичной системы счисления в шестнадцатеричную.

**Входные данные**

Программа получает на вход строку, состоящую из нулей и единиц, длина которой не превосходит 1000 символов. Первый символ строки всегда единица. Данная строка является двоичной записью некоторого числа, которое необходимо записать в шестнадцатеричном виде и вывести с использованием цифр 0, ..., 9 и букв A, ..., F без лидирующих нулей.

**Выходные данные**

Выведите результат перевода.

1. Пояснения решения

Внутри цикла происходит построение строки str4, которая содержит последние 4 символа из строки str. Для этого каждый символ добавляется в начало строки str4. Если длина строки str4 = 4 или достигнут конец str – преобразуется в десятичное число. В base будет содержаться значение в десятичной системе. Если x меньше 10, то к строке str16 добавляется символ, полученный путем прибавления значения переменной x к символу “0”. Иначе, к строке str16 добавляется символ, полученный путем прибавления значения переменной x-10 к символу “A”.

1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить работоспособность | 101000010010001111110001001000111011 | A123F123B |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <iostream>

#include <string>

using namespace std;

int main() {

string str("1010101101000011110000110100"),str4(""),str16("");

cin>>str;

for (int i=str.length()-1;i>=0;i--)

{

str4=str[i]+str4;

if (str4.length()==4 || i==0)

{

int x=0,base=1;

for (int j=str4.length()-1;j>=0;j--)

{

if (str4[j]=='1') x+=base;

base\*=2;

}

if (x<10)

{

str16=char('0'+x)+str16;

}

else

{

str16=char('A'+x-10)+str16;

}

str4.clear();

}

}

cout<<str16;

return 0;

}

### Задание 6 - Вывести значение байта побитно

1. Условие задачи.

Напишите программу, выводящую все биты 8-битного числа

**Входные данные**

Дано число A (0 ≤ A ≤ 255).

**Выходные данные**

Выведите число A в битовой форме: 8 бит, старшие биты слева, младшие – справа.

1. Пояснения решения

По аналогии с заданием 3 двигать разряды числа a на I порядков вправо, используя цикл for. Умножить (&) полученные значения на 1 и вывести результат.

1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить работоспособность | 5 | 00000101 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <iostream>

int main() {

int a;

std::cin >> a;

for (int i=7;i>=0;i--) std::cout << (a>>i & 1);

return 0;

}

### Задание 7 - Двоичные строки заданной длины

1. Условие задачи.

По данному числу N выведите все строки длины N из нулей и единиц в лексикографическом порядке.

**Входные данные**

Задано единственное число N (натуральное, 1 ≤ N ≤ 10).

**Выходные данные**

Необходимо вывести все строки длины N из нулей и единиц в лексикографическом порядке, по одной на строке.

1. Пояснения решения

Основная идея заключается в том, что каждая бинарная строка длины n может быть представлена как число от 0 до 2^n - 1. Используя это представление, можно перебирать все числа от 0 до 2^n - 1 и выводить их в бинарном формате, что и даст нам все возможные бинарные строки длины n.

1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить работоспособность | 2 | 00  01  10  11 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <iostream>

using namespace std;

int main() {

int n,m;

cin>>n;

m=1<<n;

for (int i=0;i<m;i++){

string str="";

int num=i;

for(int j=0;j<n;j++){

str=(char)((num%2)+'0')+str;

num/=2;

}

cout <<str<<endl;

}

return 0;

}

### Задание 8 - Задача на программирование: небольшое число Фибоначчи

1. Условие задачи.

Дано целое число 1≤n≤40, необходимо вычислить n-e число Фибоначчи (напомним, что F0=0, F1=1 и Fn= Fn-1+ Fт-2 , при n ≥2).

1. Пояснения решения

Просто проходимся циклом до n значения. Выводим его.

1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить работоспособность | 3 | 2 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <cassert>

#include <iostream>

class Fibonacci {

public:

static int get(int n) {

assert(n >= 0);

// put your code here

int a = 0;

int b = 1;

int i = 0;

while (i < n)

{

std::swap(a, b);

a += b;

++i;

}

return a;

}

};

int main(void) {

int n;

std::cin >> n;

std::cout << Fibonacci::get(n) << std::endl;

return 0;

}

### Задание 9 - Задача на программирование: последняя цифра большого числа Фибоначчи

1. Условие задачи.

Дано число 1≤𝑛≤107, необходимо найти последнюю цифру 𝑛-го числа Фибоначчи.

1. Пояснения решения

Теперь просто нужно вывести последнее число, используя %10

1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить работоспособность | 841645 | 5 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <cassert>

#include <iostream>

class Fibonacci final {

public:

static int get\_last\_digit(int n) {

assert(n >= 1);

int a = 0, b = 1, temp;

for (int i = 1; i < n; i++) {

temp = (a + b) % 10;

a = b;

b = temp;

}

return b;

return n;

}

};

int main(void) {

int n;

std::cin >> n;

std::cout << Fibonacci::get\_last\_digit(n) << std::endl;

return 0;

}

### Задание 10 - Задача на программирование: очередь с приоритетами

1. Условие задачи.

Первая строка входа содержит число операций 1≤𝑛≤105. Каждая из последующих *n* строк задают операцию одного из следующих двух типов:

* 𝐼𝑛𝑠𝑒𝑟𝑡 х, где 0≤𝑥≤109— целое число;
* 𝐸𝑥𝑡𝑟𝑎𝑐𝑡𝑀𝑎𝑥

Первая операция добавляет число 𝑥 в очередь с приоритетами, вторая — извлекает максимальное число и выводит его.

1. Пояснения решения

siftup(int i): Поднимает элемент на позиции i вверх по куче до тех пор, пока не будет восстановлено свойство кучи.

insert(int v): Вставляет элемент v в кучу и вызывает siftup для восстановления свойства кучи.

siftdown(int i): Опускает элемент на позиции i вниз по куче до тех пор, пока не будет восстановлено свойство кучи.

ex\_max(): Извлекает максимальный элемент из кучи, заменяя его последним элементом и вызывает siftdown для восстановления свойства кучи.

1. Таблица тестов

|  |  |  |  |
| --- | --- | --- | --- |
| Номер теста | Назначение  теста | Входные данные | Выходные данные |
| 1 | Проверить работоспособность | 2  Insert 5  any | 5 |

1. Кодирование алгоритма или запись алгоритма на языке С++

#include <bits/stdc++.h>

using namespace std;

struct Heap {

void siftup (int i) {

while (i > 0 && a[i/2] < a[i]) {

swap(a[i], a[i/2]);

i /= 2;

}

}

void insert (int v) {

n++;

a[n] = v;

siftup(n);

}

void siftdown (int i) {

while (2\*i <= n) {

int j = i;

if (a[2\*i] > a[j]) j = 2\*i;

if (2\*i+1 <= n && a[2\*i+1] > a[j]) j = 2\*i+1;

if (j == i) break;

swap(a[i], a[j]);

i = j;

}

}

int ex\_max() {

int ex\_m = a[0];

a[0] = a[n];

n--;

siftdown(0);

return ex\_m;

}

int a[100000];

int n = -1; //размер кучи

};

int main() {

Heap h;

int n, num;

string s;

cin >> n;

for (int i = 0; i < n; i++){

cin >> s;

if (s == "Insert"){

cin >> num;

h.insert(num);

}

else cout << h.ex\_max() << endl;

}

return 0;

}