Задача 1. Тежки монети

В кралството на ниската земя имало една величествена стена, която векове наред пазела територията му от нашествия. Преди точно една година обаче шайка разбойници решили, че стената е твърде сложна, и я изгорили до основи. Сега принцеса Мария е планирала да наеме опитни строители, които да възстановят стената в първоначалния ѝ блясък. За да плати на строителите, тя отишла в кралската хазна и открила $N = 2^K$ монети, всяка от които е номерирана с число от 0 до $N - 1$. Придворният счетоводител я информирал, че има два вида монети – леки и тежки, като теглото на всички тежки монети е едно и също, както и теглото на всички леки монети. Освен това според счетоводителя има поне една тежка и поне една лека монета в хазната.

Според условията, договорени със строителите на новата стена, принцеса Мария трябва да им плати само с тежки монети, като преди работата да започне, тя дължи аванс от една тежка монета. Проблемът е, че счетоводителят не знае колко и кои от монетите са тежки. За щастие той разполага с везни, с които може да сравни сумарната тежест на две множества, състоящи се от равен брой монети. Сега счетоводителят е изправен пред задачата да провери дали броят на тежките монети в кралската хазна ще бъде достатъчен за разплащането, както и да намери една такава, с която да бъде платен авансът. Това обаче би му отнело твърде много време, а принцеса Мария иска строежът на новата стена да започне колкото се може по-скоро и затова тя се обръща към Вас за помощ.

Задача

Напишете програма `coins`, която съдържа функцията *`count_heavy`*, която ще се компилира с програма на журито и, комуникирайки с нея, ще намери броя на тежките монети в кралската хазна, както и номера на една такава.

Детайли по имплементацията

Вашата функция трябва да има следния формат:

```
pair<int, int> count_heavy(int k)
```

Тя ще получава като параметър числото K (напомняме, че броят на монетите е 2^K) и трябва да върне двойка от две цели числа – броя на тежките монети и номера на една такава монета.

За комуникация с програмата на журито Ви се предоставя следната функция:

```
int weigh(const vector<int> &a, const vector<int> &b)
```

Тя приема като параметри два вектора, съдържащи номерата на монетите, които участват в претеглянето. Подадените вектори трябва да имат равен брой елементи и не трябва да съдържат повтарящи се числа. Функцията връща стойност:

- 0, ако двете множества от монети имат равно сумарно тегло;
- -1, ако теглото на монетите, чиито номера са във вектора a, е по-малко от теглото на тези, чиито номера са в b;
- $+1$, ако теглото на монетите, чиито номера са във вектора a, е по-голямо от теглото на тези, чиито номера са в b.

Обърнете внимание, че сложността за изпълнение на функцията *`weigh`* е пропорционална на сумарния брой на елементите във векторите, получени като параметри.

Вие трябва да предадете към системата файл `coins.cpp`, който съдържа функцията *`count_heavy`*. Той може да съдържа и друг код, и функции, необходими за работата Ви, но не трябва да съдържа главната функция *`main`*. Също така, не трябва да четете от стандартния вход или да пишете на стандартния изход. Програмата Ви трябва да включва хедър файла `coins.h` чрез указание към препроцесора:

```
#include "coins.h"
```

Ограничения

$1 \leq K \leq 20$

Оценяване

Ако по време на изпълнението си решението Ви даде верен отговор на даден тест, използвайки не повече от $10\,000$ извиквания на функцията `weigh`, за този тест ще бъде изчислен коефициент според формулата $coef = min\left(1, \frac{11}{10} * \left(\frac{author+1}{yours+1}\right)\right)$, където $author$ е **максималният** брой извиквания на функцията `weigh` за текущия тест на някое от предвидените авторови решения (вижте секцията за подзадачите), а $yours$ е броят на извикванията на функцията `weigh`, извършени от Вашето решение. В противен случай, за този тест $coef$ ще има стойност 0.

За всяка подзадача са предвидени определен брой групи, като всяка група е от точно три теста. Всяка от групите се оценява **независимо**, като броят точки, които ще получите за нея е равен на произведението от предвиден брой точки за нея и минималната стойност на коефициента $coef$, която се е получила за някой от трите теста в нея.

Подзадачи

Подзадача	Точки	$author$	Други ограничения
1	5	≤ 40	$K \leq 5$
2	10	≤ 250	Има точно една тежка или точно една лека монета.
3	20	≤ 1000	Има не повече от K тежки или не повече от K леки монети.
4	25	≤ 100	$K \leq 10$
5	40	≤ 500	Няма други ограничения.

Примерна комуникация

Функция на състезателя	Програма на журито
	Извиква count_heavy(2)
Извиква weigh({0, 1}, {2, 3})	Връща стойност -1
Извиква weigh({1, 2}, {0, 3})	Връща стойност +1
Извиква weigh({1}, {2})	Връща стойност 0
Връща стойност {3, 2}	

Локално тестване

За локално тестване са предоставени файловете `coins.h` и `Lgrader.cpp`. Сложете Вашия файл `coins.cpp` и двата предоставени файла в една папка. Като компилирате заедно трите файла ще получите програма, с която ще проверите верността на функцията Ви. Програмата ще изисква от стандартния вход следната последователност от данни:

- на първия ред: едно цяло положително число, задаващо стойността на K.
- на втория ред: 2^K на брой числа, със стойности 1 или 0, в зависимост от това дали съответната монета е тежка или лека.

Програмата ще отпечата информация за извикванията на функцията `weigh`, отговора, който е върнала Вашата функция `count_heavy` и съобщение дали той е верен или не.