**Задача 1**По въведени цели неотрицателни числа и , изведете -тия бит на числото .

**Задача 2**Променете -тия бит на 0 на цяло неотрицателно число.

**Задача 3**Променете -тия бит на 1 на цяло неотрицателно число.

**Задача 4**Обърнете -тия бит на цяло неотрицателно число.

**Задача 5**Обърнете младшите 5 бита на цяло неотрицателно число.

**Задача 6**Разменете стойността на две целочислени променливи използвайки само побитови операции.

*Забележка: Стандартният алгоритъм за размяна на стойностите на 2 променливи (чрез трета променлива) е почти винаги по-бърз, а освен това е и много по-разбираем и по-универсален от други начини за размяна, подобни на този от* ***задача 6****. Задачата е предвидена само за упражнение на побитовите операции, а не да покаже хубав алгоритъм за размяна на променливи.*

*Забележка 2: Може би сте виждали и други „трикове“ за размяна на стойностите на 2 променливи като например чрез събиране и изваждане или чрез умножение и деление. Освен гореизброените проблеми, тези „трикове“ имат още по-голям пробем – те не са коректни (може да получите overflow, да разделите на 0 и т.н.).*

**Задача 7**

Напишете програма програма, която чрез побитови операции проверява дали дадено цяло число е четно или нечетно.

*Съвет: Подобно на предходната задача, идеята на тази задача е да се добие представа за побитовите операции. Счита се за по-добра практика подобни проверки да се правят по по-интуитивен начин (в случая чрез деление), тъй като в противен случай кодът става трудно четим. Въпреки, че делението е в (доста) пъти по-бавна операция от побитовите операции, почти всеки модерен компилатор би оптимизирал подобен код, така че в генерираната програма да не участва инструкция за деление.* [*Live example*](https://godbolt.org/#g:!((g:!((g:!((h:codeEditor,i:(filename:'1',fontScale:12,fontUsePx:'0',j:1,lang:___c,selection:(endColumn:2,endLineNumber:9,positionColumn:2,positionLineNumber:9,selectionStartColumn:2,selectionStartLineNumber:9,startColumn:2,startLineNumber:9),source:'int+isOdd_division(int+num)%0A%7B%0A++++return+num+%25+2+!!%3D+0%3B%0A%7D%0A%0Aint+isEven_bitwise(int+num)%0A%7B%0A++++return+num+%26+1%3B%0A%7D'),l:'5',n:'1',o:'C+source+%231',t:'0')),k:50.5248759136807,l:'4',n:'0',o:'',s:0,t:'0'),(g:!((g:!((h:compiler,i:(compiler:cg142,filters:(b:'0',binary:'1',binaryObject:'1',commentOnly:'0',debugCalls:'1',demangle:'0',directives:'0',execute:'1',intel:'0',libraryCode:'0',trim:'1',verboseDemangling:'0'),flagsViewOpen:'1',fontScale:12,fontUsePx:'0',j:2,lang:___c,libs:!(),options:'-O3+-std%3Dc11',overrides:!(),selection:(endColumn:12,endLineNumber:8,positionColumn:12,positionLineNumber:8,selectionStartColumn:12,selectionStartLineNumber:8,startColumn:12,startLineNumber:8),source:1),l:'5',n:'0',o:'+x86-64+gcc+14.2+(Editor+%231)',t:'0')),k:32.99078049509948,l:'4',m:50,n:'0',o:'',s:0,t:'0'),(g:!((h:output,i:(compilerName:'x86-64+clang+15.0.0',editorid:1,fontScale:14,fontUsePx:'0',j:2,wrap:'1'),l:'5',n:'0',o:'Output+of+x86-64+gcc+14.2+(Compiler+%232)',t:'0')),header:(),l:'4',m:50,n:'0',o:'',s:0,t:'0')),k:49.4751240863193,l:'3',n:'0',o:'',t:'0')),l:'2',n:'0',o:'',t:'0')),version:4)

**Задача 8**

Напишете програма, която по дадено цяло неотрицателно число пресмята .

**Задача 9**

Съобщение от 3 символа е кодирано както следва:

* най-младшите 2 бита на кода на първия символ са преместени, така че да станат най-старши;
* местата на най-старшия и най-младшия бит на кода на втория символ са разменени;
* всеки от средните 4 бита на кода на третия символ е обърнат.

Напишете програма, която въвежда 3 цели числа – кодовете след като е било приложено описаното кодиране. Програмата да декодира съобщението и да изведе трите оригинални символа.

**Пример**

**Вход**

216 224 72

**Възможен изход**

Cat

**Задача 10**

Решете **задача 2** от [тази](https://docs.google.com/document/d/1Q0DLRYvm6R9a-crCPndrsHRJD6xk-mou4RrRLcqKdWE/edit?tab=t.0) тема от домашно от предна година. (Разбира се, може да решите и останалите задачи :))

**Задача 11\* (**[**The game of nim**](https://en.wikipedia.org/wiki/Nim)**)**

Имаме купчета с камъчета. Alice и Bob се редуват да правят ходове, като на всеки ход премахват произволен брой камъчета от произволно купче. Всеки от тях е длъжен, когато е на ход, да премахме поне едно камъче и освен това не е позволено за един ход да премахват камъчета от различни купчета. Играчът който първи не може да направи валиден ход губи играта. Съответно другият играч печели играта. Alice и Bob са перфектни математици и винаги играят по най-добрия възможен начин.

Вашата задача е да напишете програма, която определя кой ще спечели играта. На стандартния вход е дадено числото – броят купчета. Следват числа – броят камъчета в поредното купче. Най-накрая се въвежда кой от играчите е на ход – съответно числото 1 ако Alice е на ход и числото 2 ако Bob е на ход.

На стандартния изход трябва да изведете кой от двамата играчи ще спечели играта. Ако играчът който е на ход печели, изведете и произволен негов печеливш ход – номер на купче от което се взима (броено от 1) и брой камъчета които да бъдат взети.

**Пример**

**Вход**3  
1 5 3  
2  
**Възможен изход**2  
2 3**Вход**3  
1 2 3  
1  
**Изход**2

*Съвет: Ако не сте запознати с цикли и масиви, можете да решите задачата за някое фиксирано , например за .*