# Apartat 1

En aquest apartat s’ha disenyat per tal de comprovar si un array de enters esta format unicament per zeros o uns.

Per implementar aquesta funció necesitem recorrer cada una de les posicions de l’array i comprovar per cada una si es un 0 o be un 1, per fer aixo he implementat una funció auxiliar anomenada isBit per tal de realitzar la comprovacio de valor correcte, a partir d’aqui unicament s’ha de buscar un valor que no compleixi amb la condició, si el trobem retornarem fals i si no el trobem i per tant sortim del for retornarem true.

# Apartat 2

En aquest apartat es demana una funció que a partir d’un string que teoricament conte un nombre binari amb els bits mes significatius a la part esquerra de l’string el transformi en un array d’enters amb els bits mes significatius a la dreta de l’array, tambe es demana que si l’string conte algun valor no binari l’ovbiem i si tant l’array com l’string son de diferents mides que trunqui o que allargui el valor segons correspongui.

Per desenvolupar aquesta funció primerament hem de tenir en compte que l’array i l’string estaran en sentits oposats, per tant he utilitzat fromPos per recorrer l’string i toPos per recorrer l’array, els quals començaran a from.length-1 i a 0 respectivament.

Tambe hem de tenir en compte que l’string esta formada per caracters i l’array ha d’estar format per enters, per tant hem d’aplicar-li una transformació al caracter en questió per tal de convertir-lo en l’enter corresponent, aixo s’aconsegueix amb la instrucció *num=from.charAt(fromPos)-'0';* .

Seguidament hem de comprovar tambe que sigui un valor binari valid, aixo ho aconseguint utilitzant la funció desenvolupada en l’apartat anterior anomenada isBit.

# Apartat 3

En aquest apartat es requereix una funció que escursi un array d’enters que conté un nombre binari, per obtenir un array de la llargada especificada, concretament menor a la actual.

Per disenyar aquesta funció primerament hem de tenir en compte si estem fent una crida valida, es a dir comprovar que la llargada desitjada es mes gran o igual a 0 i que es mes petita o igual que la del array d’origen, un cop comprobat aixo simplement podem utilitzar la funcio copy desenvolupada en l’apartat anterior ja que si els dos arrays son de la mateixa mida simplement copiara de l’un a laltre i si la llargada del destí es mes petita que la de l’origenn truncarà.

# Apartat 4

La funció a realitzar ha de permetre allargar un nobre binari contingut en un array d’enters a una llargada especificada, concretament mes gan o igual a la actual, tenint en compte el signe del valor original, ja que en cas de ser positiu s’ha d’afegir 0 i en cas de ser negatiu s’ha d’afer 1 en els bits de mes pes per tal de mantenir el valor numeric.

Per resoldre aquest aprtat, que resulta molt semblant a l’anterior primerament comprovem que la llargada desitjada es mes gran o igual a 0 i tambe mes gran o igual a la actual, un cop comprovat aixo, comprovem si es negatiu el nombre mirant siel bit de mes per es igual a 1, si es aixi creem un array de la mesura desitjada pero ple de uns, utilitzant la funció auxiliar oneOfSize, en cas contrari el creem ple de zeros, utilitzant la funció zeroOfSize, i simplement ens queda copiar els valors ja existents cosa que realitzem mitjançant la funció copy desenvolupada anteriorment i retornem el resultat.

# Apartat 5e

# Apartat 6

# Apartat 7

# Apartat 8

# Apartat 9

# Apartat 10