**2.1** (a) – (g).

**(a)** “isBrotherOf” on the set of people.

Symmetric: False, sister can’t be brother of

Reflexive: False, can’t be a brother to yourself

Transitive: True, 1 is brother of 2 and 2 is brother of 3 so 1 is brother of 3

Anti-symmetric: False, the two are not the same person

**(b)** “isFatherOf” on the set of people

Symmetric: False, can’t be a father to self

Reflexive: False, child can’t be father of Father

Transitive: False, 2 generations back is grand father

Anti-symmetric: True, because father can’t go both ways

**(c)** The relation for real numbers x and y.

Symmetric: False, 1 is the only one in the relation, (2,2) = 8 not 1

Reflexive: False, 1 is the only one in the relation, (2,2) = 8 not 1

Transitive: False, ( .4 , .916 ) -> ( .916 ,.4 ). The .4 isn't equal to .4

Anti-symmetric: False, ( .5 , .707 ) is not the same as ( .707 , .5 )

**(d)** The relation for real numbers x and y.

Symmetric: true,

Reflexive: True, a = b and b = a

Transitive: True,

Anti-symmetric: False because

**(e)** The relation for

Symmetric: False, 1 mod 2 != 0

Reflexive: True, 1 mod 1 is zero and is same for the rest of the set

Transitive: True, they all equal 0

Anti-symmetric: True,1 mod 2 != 0

**(f)** The empty relation∅(i.e., the relation with no ordered pairs for which

Symmetric: True, you don't have (a,b) so you don't need (b,a)

Reflexive: False, everything isn’t related to itself

Transitive: True, you don't have (a,b) so don’t need (b,c) for (a,c)

Anti-symmetric: True, its not reflexive

**2.2** (a) – (f).

For each of the following relations, either prove that it is an equivalence relation or prove that it is not an equivalence relation.

**(a)** For integers a and b, a ≡ b if and only if a + b is even.

This is not an equivalence relation because 1+2 is an odd number

**(b)** For integers a and b, a ≡ b if and only if a + b is odd.

Is not an equivalence relation because reflexive case fails 1+1 = 2 <--even

**(c)** For nonzero rational numbers a and b, a ≡ b if and only if a×b > 0.

This is an equivalence relation because its reflexive(1,1)>0, symmetric(1,2)(2,1)>0, and transitive(1,2)(2,3)(1,3)>0

**(d)** For nonzero rational numbers a and b, a ≡ b if and only if a/b is an integer.

Is not an equivalence relation because symmetric fails (b,a) is fraction

**(e)** For rational numbers a and b, a ≡ b if and only if a−b is an integer.

This is an equivalence relation because its reflexive( 1-1=0 , symmetric( 1-2=-1,2-1=1), and transitive(1-2=-1 , 2-3=-1 , 1-3=-2)

**(f)** For rational numbers a and b, a ≡ b if and only if|a−b|≤ 2.

Is not an equivalence relation because it is not transitive

**2.3** (a) – (f).

State whether each of the following relations is a partial ordering, and explain why or why not.

**(a)** “isFatherOf” on the set of people.

Is not a partial order because a != c for transitive

**(b)** “isAncestorOf” on the set of people.

Is a partial order because antisymmetric case the (b,a) isnt there and (a,b)(b,c)->(a,c)

**(c)** “isOlderThan” on the set of people.

Is a partial order because antisymmetric case the (b,a) isnt there and (a,b)(b,c)->(a,c)

**(d)** “isSisterOf” on the set of people.

Is not a partial order because a != c for transitive

**(e)** {ha,bi,ha,ai,hb,ai}on the set{a,b}.

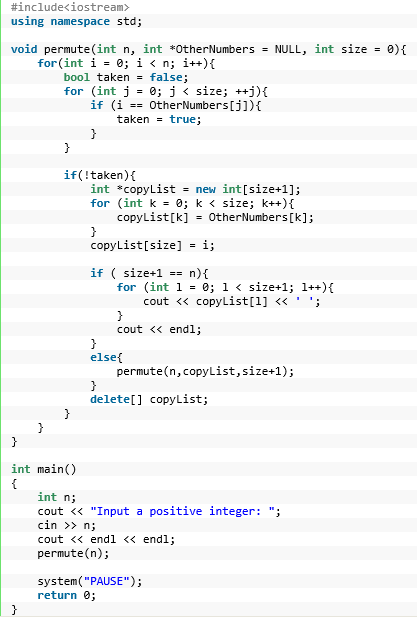
not a partial order because its not transitive

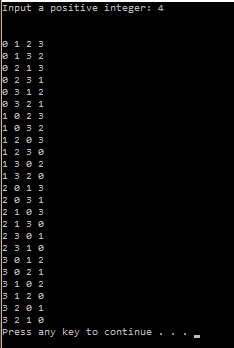
**(f)** {h2,1i,h1,3i,h2,3i}on the set{1,2,3}.

not a partial order because its not transitive

**2.15**. Your function should have the prototype void perms(int A[], int n); Test your function separately and include in your answer what the output would be if you called your function with arguments int A[4] = {1,2,3,4}, n = 4.

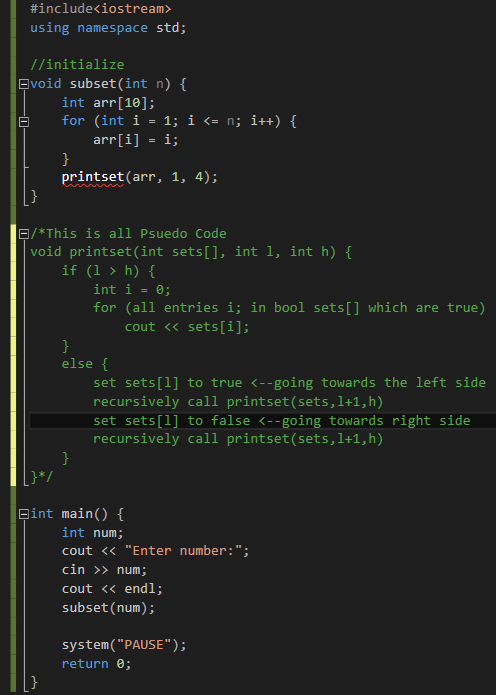
Write a function to print all of the permutations for the elements of an array containing n distinct integer values





**2.16**. Your function should have the prototype void subsets(int n); Subsets should be printed, for example, as {}, {1}, {1,2}, etc.

Write a recursive algorithm to print all of the subsets for the set of the ﬁrst n positive integers



**2.22** – **2.24**.

**22)** Prove Equation 2.2 using mathematical induction.

**2.2)**

Left side = 12 = 1 | right side =

Both sides are equal so p(1) is true

Now p(n)

1 2 + 2 2 + 3 2 + ... + n 2 = now, p(n+1)

1 2 + 2 2 + 3 2 + ... + n 2 + (n + 1) 2 =

=

=

= -> which is the same as the

statement for p(n+1)

**23)** Prove Equation 2.6 using mathematical induction.

**2.6)**

**24)** Prove Equation 2.7 using mathematical induction.

**2.7)**

**2.28**.

Prove that .

Base cases: n=1: Fib(1) = 1, and > 1

n=2: Fib(2) =1 < .

Assume , where n > 2.

Since