**PROBLEM :**

Design a program to obtain the relation as the ordered pairs given its hasse diagram.

**Table of contents**

|  |  |  |
| --- | --- | --- |
| SL.NO. | TITLE | PAGE NO. |
| 1 | ABSTRACT |  |
| 2 | INTRODUCTION |  |
| 3 | ANALYTICAL SOLUTION |  |
| 4 | FLOWCHART |  |
| 5 | ALGORITHM |  |
| 6 | PROGRAM |  |
| 7 | OUTPUT |  |
| 8 | APPLICATIONS |  |
| 9 | CONCLUSION |  |
| 10 | BIBLIOGRAPHY |  |

**Abstract**

The objective of the given program is two folds. To obtain relation as ordered pairs in the given problem we will have to determine the following

1. Taking input as poset elements & finding its hasse diagram.

2. Then obtaining the relation as the ordered pairs through the hasse diagram.

The above two objectives are achieved with the help of the program in the following ways

**Procedure used-**

1. Input the elements in 1-D array and obtain its hasse diagram by checking whether the number of elements entered is even or odd.

2. Then displaying the relation as ordered pairs.

**Input method-**

First of all, user must enter the number of elements in the poset and then its elements, which will be stored in 1-D array. Here we have restricted the user to enter only set of all positive integers. User must enter the elements of poset in ascending order to obtain a properly designed hasse diagram.

**Method to obtain Hasse diagram-**

In general case, hasse diagram is obtained by determining whether the number of elements entered in the poset is even or odd & in particular cases, hasse diagram of the elements is obtained by entering the value of the positions to the built-in hasse diagram.

**Output method-**

Here we have considered the divisibility relation on set of all positive integers. If one of the elements divides the other completely then those two elements form an ordered pair & all such pairs form the output relation.

**INTRODUCTION**

**POSET-**

A (non-strict) partial order is a [binary relation](https://en.wikipedia.org/wiki/Binary_relation) ≤ over a [set](https://en.wikipedia.org/wiki/Set_(mathematics)) P which is [reflexive](https://en.wikipedia.org/wiki/Reflexive_relation), [antisymmetric](https://en.wikipedia.org/wiki/Antisymmetric_relation), and [transitive](https://en.wikipedia.org/wiki/Transitive_relation), i.e., which satisfies for all a, b, and c in P:

* a ≤ a ([reflexivity](https://en.wikipedia.org/wiki/Reflexive_relation): every element is related to itself).
* if a ≤ b and b ≤ a, then a = b ([antisymmetry](https://en.wikipedia.org/wiki/Antisymmetric_relation): there exists at most one relation between two distinct elements)
* If a ≤ b and b ≤ c, then a ≤ c ([transitivity](https://en.wikipedia.org/wiki/Transitive_relation): if a first element is related to a second element, and, in turn, that element is related to a third element, then the first element is related to the third element).

In other words, a partial order is an antisymmetric preorder.

A set with a partial order is called a partially ordered set (also called a poset).

**RELATION-**

|  |
| --- |
| Relation:  A relation is simply a set of ordered pairs. |

  The first elements in the ordered pairs (the x-values), form the domain.  The second elements in the ordered pairs (the y-values), form the range.  Only the elements "used" by the relation constitute the range.

|  |  |  |
| --- | --- | --- |
| http://www.regentsprep.org/regents/math/algtrig/atp5/relationsets.gif | This mapping shows a relation from set A into set B. This relation consists of the ordered pairs, (1,2), (3,2), (5,7), and (9,8).   |  | | --- | | •  The domain is the set {1, 3, 5, 9}. •  The range is the set {2, 7, 8 }.    (Notice that 3, 5 and 6 are not part of the range.) •  The range is the dependent variable. | |

**HASSE DIAGRAM-**

In [order theory](https://en.wikipedia.org/wiki/Order_theory), a Hasse diagram is a type of [mathematical diagram](https://en.wikipedia.org/wiki/Mathematical_diagram) used to represent a finite [partially ordered set](https://en.wikipedia.org/wiki/Partially_ordered_set), in the form of a [drawing](https://en.wikipedia.org/wiki/Graph_drawing) of its [transitive reduction](https://en.wikipedia.org/wiki/Transitive_reduction). Concretely, for a partially ordered set (S, ≤) one represents each element of S as a [vertex](https://en.wikipedia.org/wiki/Vertex_(graph_theory)) in the plane and draws a [line segment](https://en.wikipedia.org/wiki/Line_segment) or curve that goes upward from x to y whenever y [covers](https://en.wikipedia.org/wiki/Covering_relation) x (that is, whenever x < y and there is no z such that x < z < y). These curves may cross each other but must not touch any vertices other than their endpoints. Such a diagram, with labelled vertices, uniquely determines its partial order.

Hasse diagrams are named after [Helmut Hasse](https://en.wikipedia.org/wiki/Helmut_Hasse) (1898–1979); according to [Birkhoff (1948)](https://en.wikipedia.org/wiki/Hasse_diagram#CITEREFBirkhoff1948), they are so called because of the effective use Hasse made of them.

The phrase "Hasse diagram" may also refer to the transitive reduction as an abstract [directed acyclic graph](https://en.wikipedia.org/wiki/Directed_acyclic_graph), independently of any drawing of that graph, but this usage is eschewed here.

Although Hasse diagrams are simple as well as intuitive tools for dealing with finite [posets](https://en.wikipedia.org/wiki/Partially_ordered_set), it turns out to be rather difficult to draw "good" diagrams. The reason is that there will in general be many possible ways to draw a Hasse diagram for a given poset. The simple technique of just starting with the [minimal elements](https://en.wikipedia.org/wiki/Minimal_element) of an order and then drawing greater elements incrementally often produces quite poor results: symmetries and internal structure of the order are easily lost.

**Analytical solution-**

Let the elements in the poset be 2, 4,8,16. Now as the condition is divisibility relation on set of all positive integers, we have to check whether x divides y to obtain relation as ordered pairs from hasse diagram.

Consider, (2, 4) => 4 is divided by 2 which is true

(2, 8) => 8 is divided by 2 which is true

(2, 16) => 16 is divided by 2 which is true

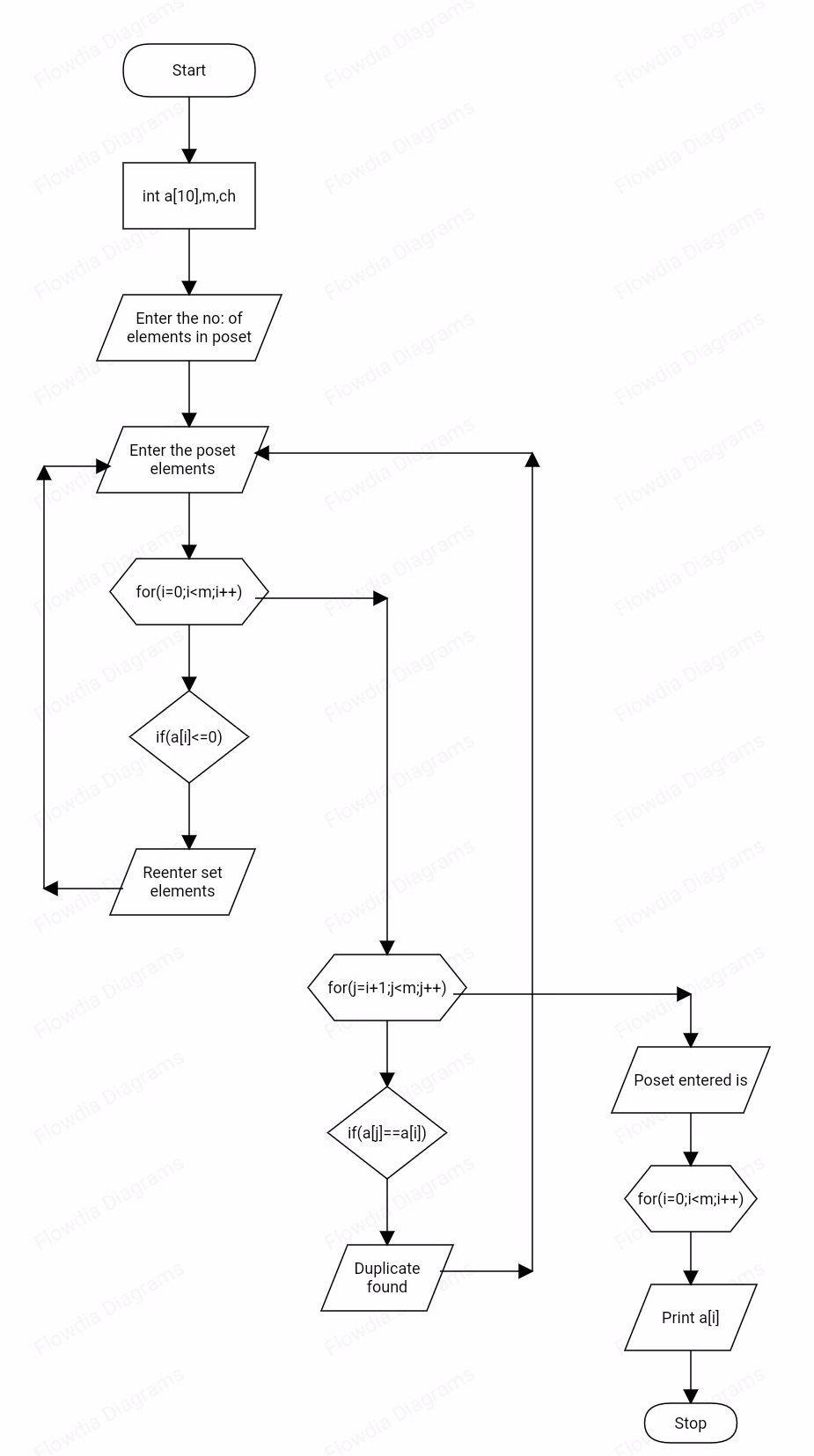
(4, 8) => 8 is divided by 4 which is true

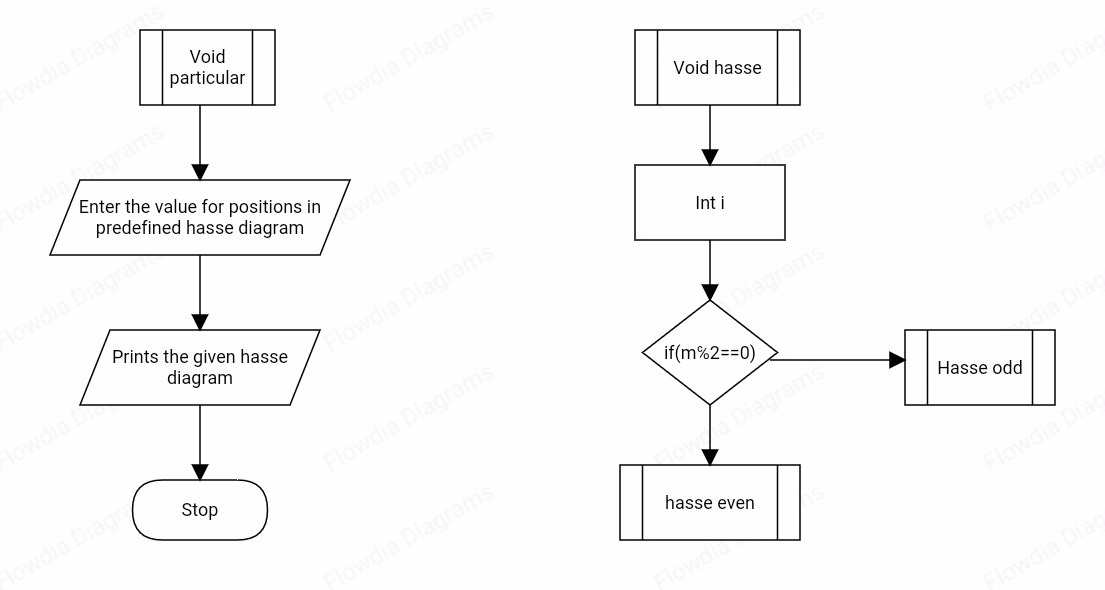
(4, 16) => 16 is divided by 4 which is true

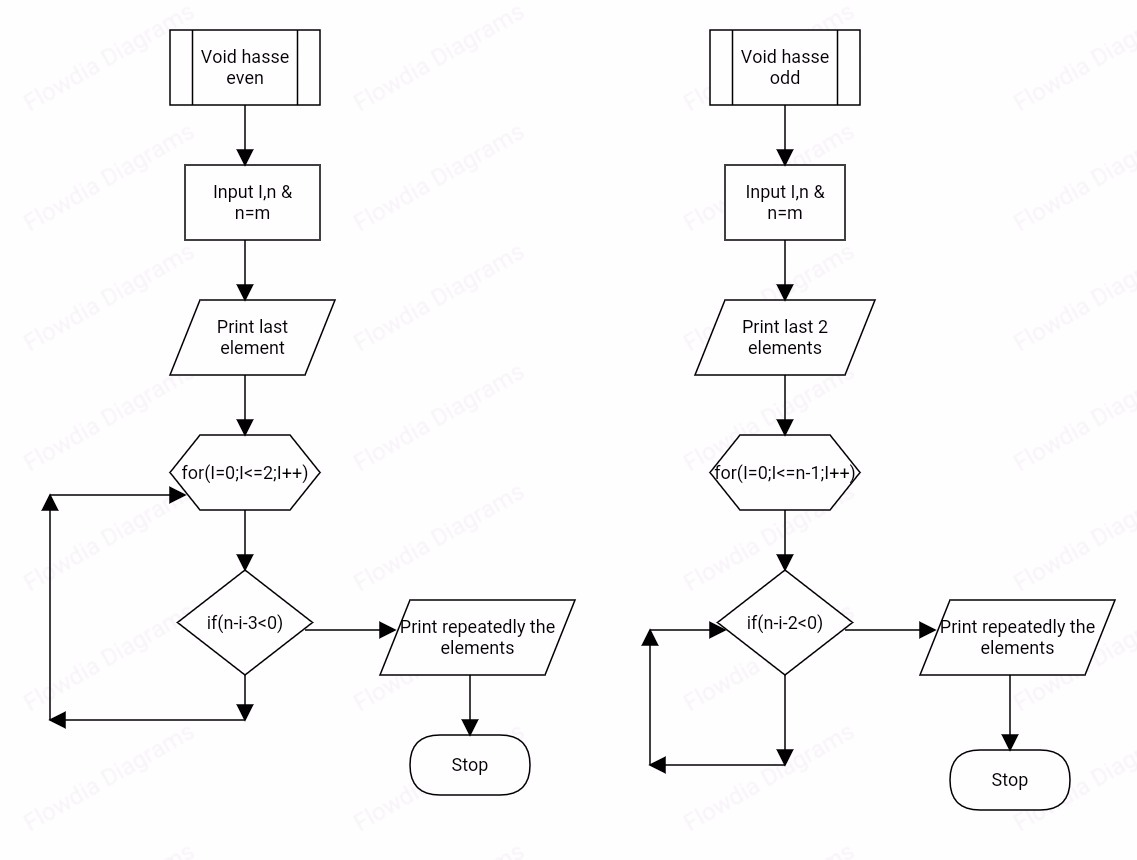
(8, 16) => 16 is divided by 8 which is true

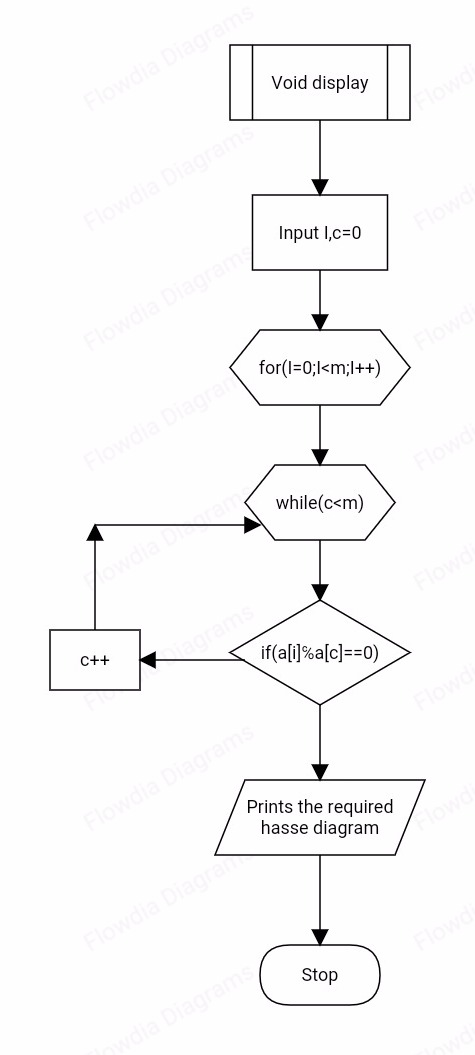
Hence the relation includes ordered pairs (2,4),(2,8),(2,16),(4,8),(4,16),(8,16)

Therefore R = { (2,4),(2,8),(2,16),(4,8),(4,16),(8,16) }

**Flowchart-**

****

****

****

**Algorithm-**

1. INPUT THE NUMBER OF ELEMENTS IN PARTIAL ORDER SET.

2. DOMAIN IS THE SET OF ALL POSITIVE INTEGERS.

3. INPUT THE POSET ELEMENTS IN ASCENDING ORDER.

4. CHECK FOR DUPLICATE ELEMENTS IN POSET IF PRESENT THEN,

-> REPEAT STEP 3

5. CHECK FOR ELEMETS OF POSET LESS THAN OR EQUAL TO ZERO (0) THEN,

-> REPEAT STEP 2

6. THE ELEMENTS OF THE POSET WILL BE PRINTED.

7. FOR GENERAL CASE GO TO EITHER STEP 8 OR 9 DEPENDING ON THE NUMBER OF ELEMENTS.

8. IF NUMBER OF ELEMENTS IN THE POSET IS EVEN THEN,

a) PRINT THE LAST ELEMENT OF THE POSET AT THE TOP.

b) RUN A LOOP FROM FIRST ELEMENT TO LAST BUT ONE ELEMENT.

// for (i=0;i<=n-2;i++)

c) REPEATEDLY PRINT ALL THE ELEMENTS WHICH ARE DIVISIBLE BY THE TOP ELEMENT.

d) WHEN n-i-3 TURNS OUT TO BE LESS THAN 0 , REPEAT STEP 8(b).

e) THE REQUIRED HASSE DIAGRAM IS DISPLAYED.

9. IF NUMBER OF ELEMENTS IN THE POSET IS ODD THEN,

a) PRINT THE LAST TWO ELEMENTS OF THE POSET AT THE TOP.

b) RUN A LOOP FROM FIRST ELEMENT TO LAST BUT ONE ELEMENT.

// for (i=0; i<=n-1; i++)

c) REPEATEDLY PRINT ALL THE ELEMENTS WHICH ARE DIVISIBLE BY THE TOP ELEMENTS.

d) IF (n-i-2) TURNS OUT TO BE LESS THAN 0, REPEAT STEP 9(b).

e) THE REQUIRED HASSE DIAGRAM IS DISPLAYED.

10. REPEATEDLY CHECK FOR THE DIVISIBILITY FROM FIRST ELEMENT WITH ALL THE SUCCESSIVE ELEMENTS OF THE POSET.

a) PRINT THE CORRESPONDING ELEMENTS AS THE ORDERED PAIRS FROM STEP (10).

11. FOR THE PARTICUALAR CASE THEN,

a) PRE DEFINED HASSE DIAGRAM IS DISPLAYED.

b) ENTER THE ELEMENTS OF THE POSET AT THE CORRESPONDING POSITIONS AS SPECIFIED IN THE PREDEFINED HASSE DIAGRAM.

c) THE CORRESPONDING HASSE DIAGRAM AS WELL AS POSET WILL BE DISPLAYED.

d) REPEATEDLY STEP 10 WILL BE EXECUTED.

12. STOP.

**Program-**

**Source code**

#include<stdio.h>

#include<stdlib.h>

void input(int a[],int m)

{

int i,j,k,c;

abc:printf("\nENTER THE PARTIAL ORDER SET ELEMENTS:\n");

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

{

scanf("%d",&a[i]);

if(a[i]<=0)

{

printf("\n DOMAIN IS SET OF ALL POSITIVE INTEGERS\n");

printf("\n RE ENTER THE SET ELEMENTS\n");

goto abc;

}

}

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

{

for(j=i+1;j<m;j++)

{

if(a[j]==a[i])

{

printf("\nDUPLICATE FOUND\nPLEASE RE ENTER THE SET ELEMENTS\n");

goto abc;

}

}

}

printf("\n PO SET ENTERED IS==>\n");

printf("\n A={");

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

{

printf("%d ",a[i]);

if(i==m-1)

break;

else

printf(",");

}

printf(" }");

}

void particular()

{

int pos[]={1,2,3,4,5,6},a[5],i;

printf("\n\n\t\tPREDEFINED HASSE DIAGRAM IS :\n");

hasee1(pos);

printf("\n\n\t\tENTER THE VALUE FOR POSITIONS : \n");

for(i=0;i<6;i++)

{

printf("\n\t\tVALUE FOR POSITION %d : ",i+1);

scanf("%d",&a[i]);

}

printf("\n\t\tTHE GIVEN HASSE DIAGRAM IS :");

hasee1(a);

printf("\n\n");

display(a,6);

}

void hasee1(int \*pos)

{

printf("\n\n\n\t\t%d<-------",pos[5]);

printf("\n\t\t^\t^\n\t\t|\t|\n\t\t%d\t%d\n\t\t^\t^\n\t\t|\t|\n\t\t%d\t%d\n\t\t^\t^\n\t\t|\t|\n\t\t%d------->",pos[3],pos[4],pos[1],pos[2],pos[0]);

}

void hasseeven(int a[],int m)

{

int i;

int n;

n=m;

printf("\t%d\n",a[n-1]);

for(i=0;i<=n-2;i++)

{

if(n-i-3<0)

break;

else

{

printf("\n\n\n%d\t\t%d\n\n\n",a[n-i-3],a[n-i-2]);

n=n-1;

}

}

printf("\t%d\n",a[0]);

}

void hasseodd(int a[],int m)

{

int i;

int n;

n=m;

for(i=0;i<=n-1;i++)

{

if(n-i-2<0)

break;

else

{

printf("\n\n\n%d\t\t%d\n\n\n",a[n-i-2],a[n-i-1]);

n=n-1;

}

}

printf("\t%d\n",a[0]);

}

void hasse(int a[],int m)

{

int i;

if(m%2==0)

hasseeven(a,m);

else

hasseodd(a,m);

}

void display(int a[],int m)

{

int c,i;

printf("\nTHE REQUIRED PARTIAL ORDER RELATION IS =====>>:\n");

printf("\nR={");

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

{

c=0;

while(c<m)

{

if(a[i]%a[c]==0)

{

printf("(%d,%d)",a[c],a[i]);

}

c++;

}

}

printf("}");

printf("\n");

}

main()

{

int a[20],m,ch;

printf("\n\nDMS EVENT 2\n\n");

printf("\nTEAM MEMBERS:\n 1:AJEYA B JOIS\n2.AJAY KB\n3.NITHIN PRABHU\n");

printf("\n($) PROBLEM : GIVEN HASSE DIAGRAM OBTAIN THE PARTIAL ORDER RELATION:\n");

printf("\n($) WE HAVE CONSIDERED THE DIVISIBILTY RELATION ON SET OF ALL POSITIVE INTEGERS FOR OUR PROBLEM\n");

printf("\n($) ie R={(x,y <<Z+) | x divides y}\n");

printf("\n($) USERS MUST ASSUME THE EDGES IN HASSE DIAGRAM IN BETWEEN THE NODES\n");

printf("\n($) USER MUST ENTER THE ELEMENTS OF POSET IN ASCENDING ORDER ONLY\n");

printf("\n($) OR ELSE HASSE DIAGRAM THAT WE OBTAIN WILL BE INCORRECT\n");

printf("\n($) USER MUST GIVE THE POSET BEFORE PROCEEDING FURTHER\n");

printf("\nENTER THE NUMBER OF ELEMENTS YOU REQUIRE IN THE POSET:\n");

scanf("%d",&m);

while(1)

{

printf("\n1.INPUT THE POSET\n2.DISPLAY HASSE DIAGRAM\n3.DISPLAY PARTIAL ORDER RELATION\n4.EXIT\n5.PARTICULAR PROBLEM\n");

printf("\nENTER YOUR CHOICE:\n\n");

scanf("%d",&ch);

switch(ch)

{

case 1:input(a,m);

break;

case 2:hasse(a,m);

break;

case 3:display(a,m);

break;

case 4:exit(0);

case 5:particular();

break;

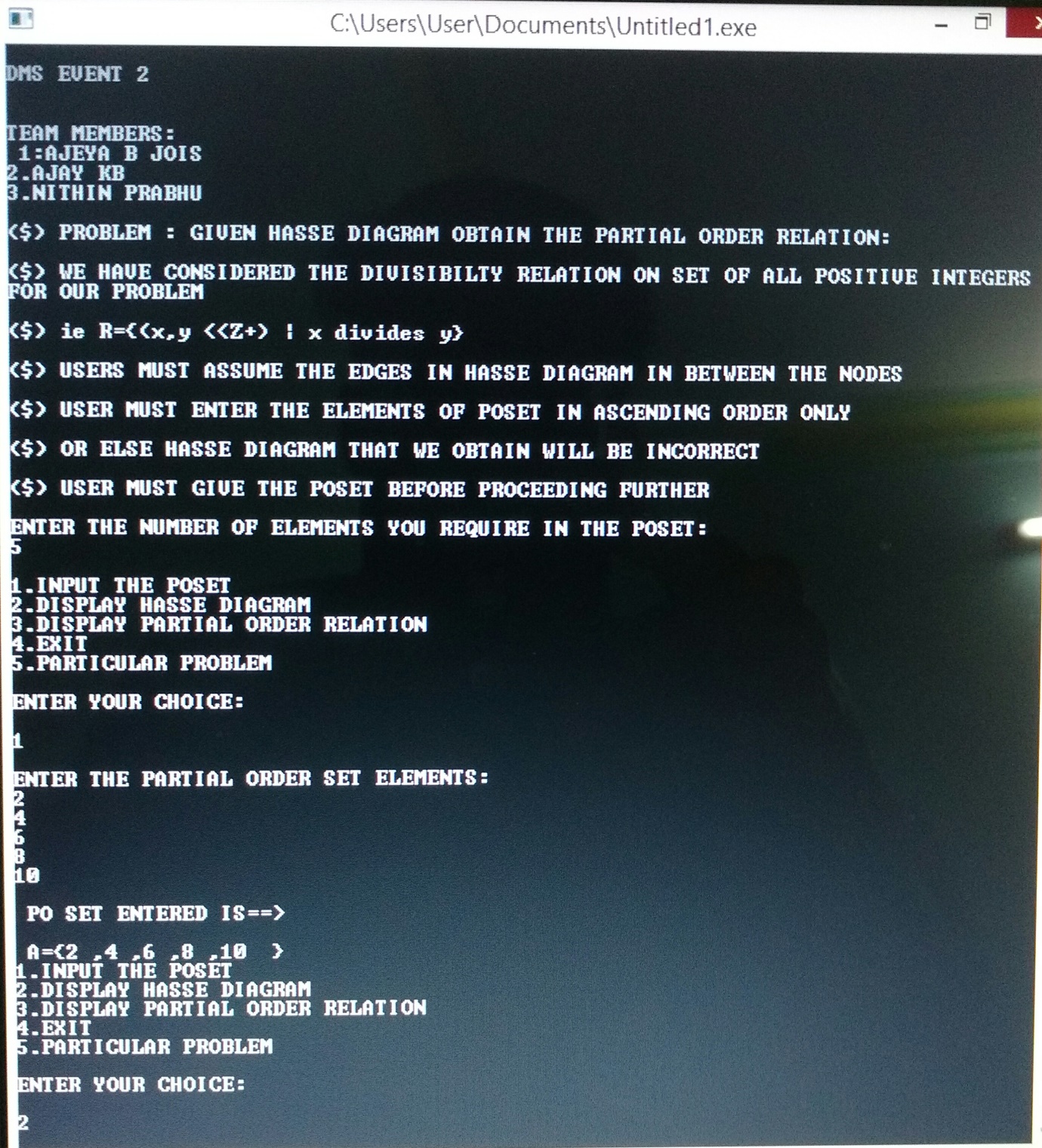
default:printf("\nINVALID CHOICE:\n");

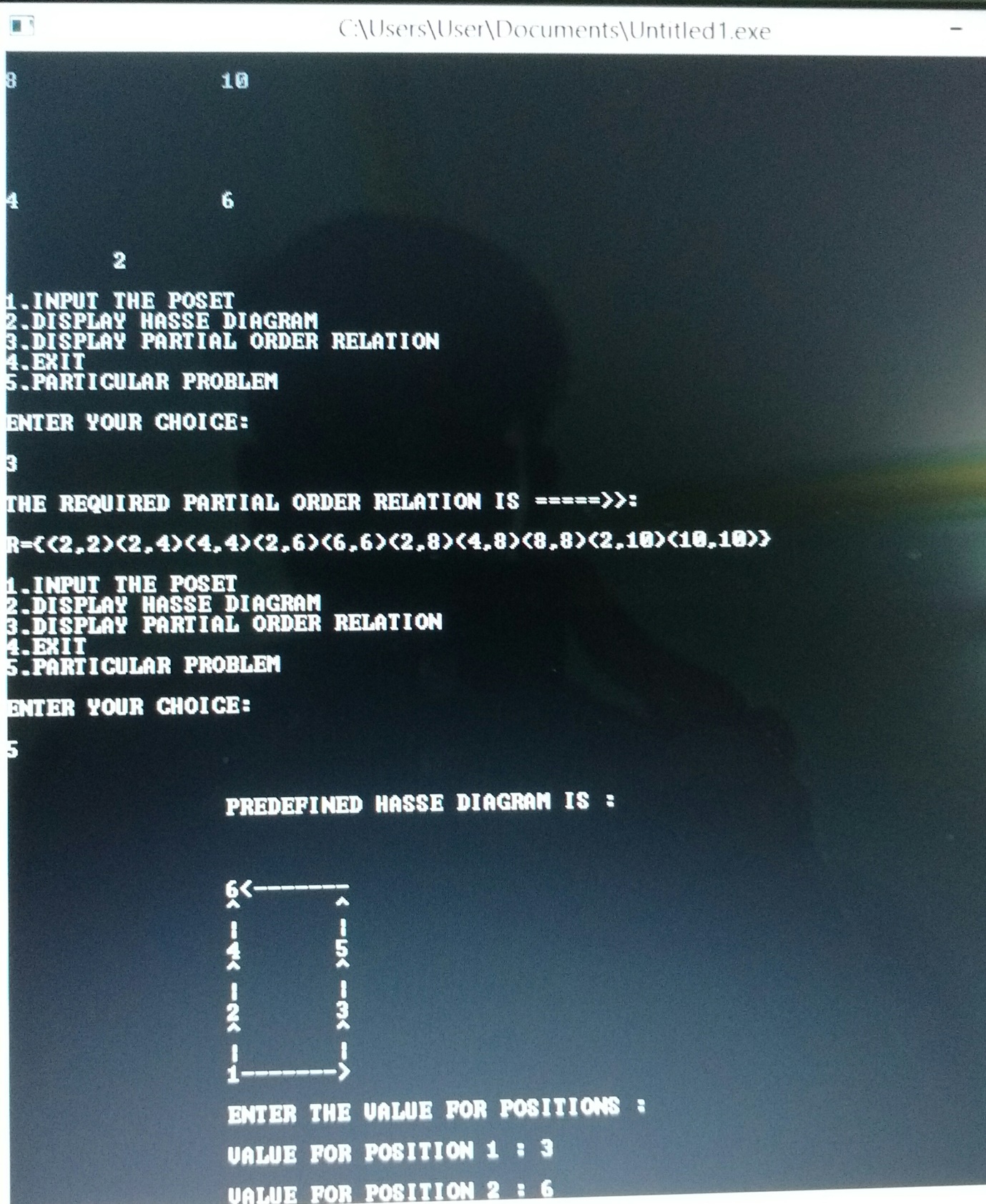
}

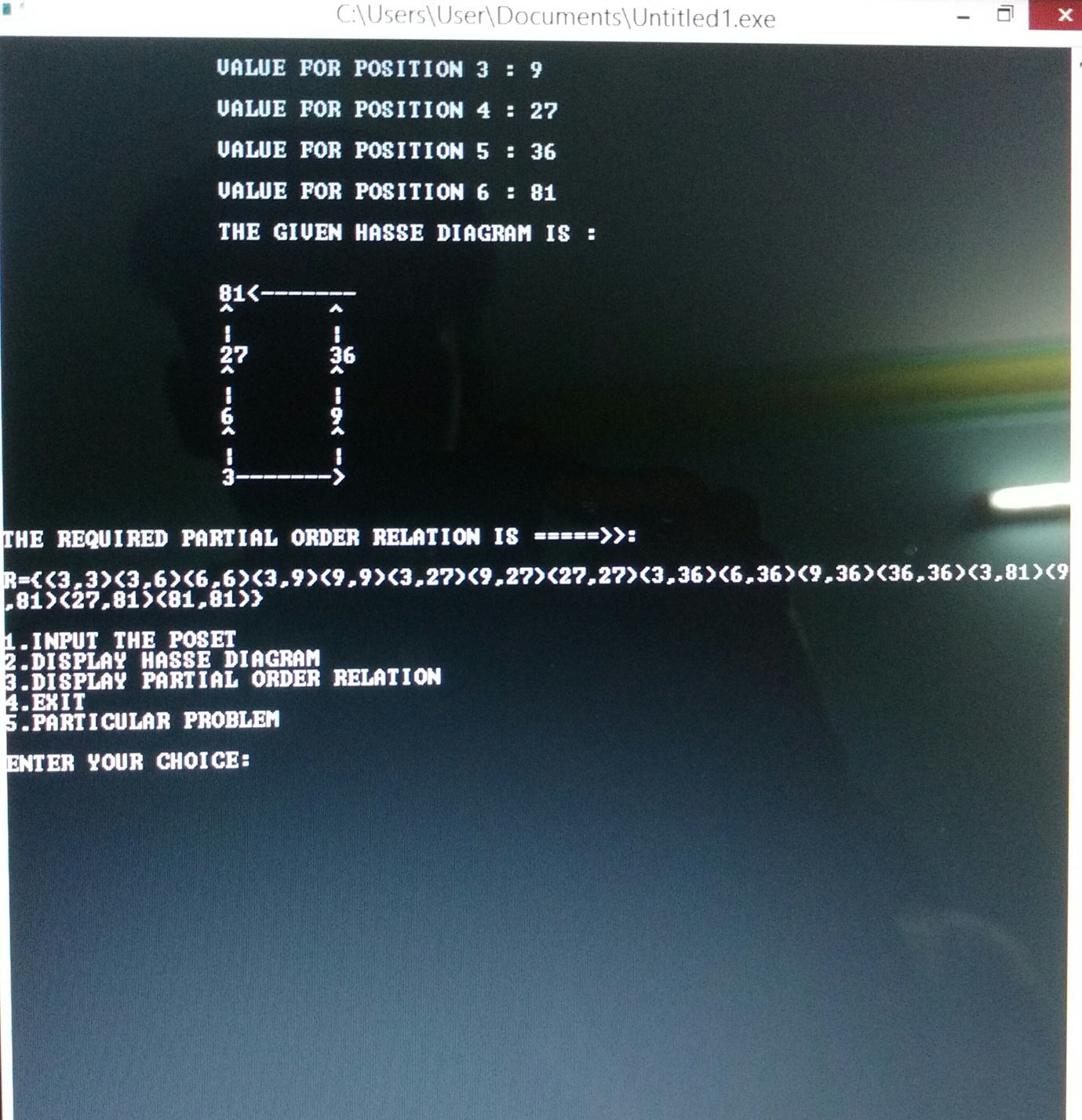
}

}

**Output-**







**Applications-**

1. In compiler design.
2. In ring theory the Hasse diagram of ideals ordered by inclusion is used often.
3. In particular the attached Moebius function is used to compute the so-called homogenous weight in Coding Theory.
4. The Jordan-Holder theorem of group theory is essentially a theorem about paths in the Hasse diagram of subgroups.
5. Hotspot prioritization system.

**CONCLUSION-**

The relation as the ordered pairs is obtained from its hasse diagram by considering the divisibility relation on set of all positive integers.

**Bibliography**

The project is a combined efforts of the below sources

**WEB SOURCES**

1. Wikipedia
2. WolFramAlpha.com
3. Stack exchange
4. Quora

**OFFLINE SOURCES**

1. Discrete and Combinational Mathematics by Grimaldi.

2. Hand book of Discrete and Combinational Mathematics by Keneth H Rosen.