**PROBLEM :**

Design a program to construct the rook’s polynomial.

**Table of contents**

|  |  |  |
| --- | --- | --- |
| SL.NO. | TITLE | PAGE NO. |
| 1 | ALGORITHM | 03 |
| 2 | PROGRAM | 06 |
| 3 | OUTPUT | 12 |
| 4 | APPLICATIONS | 15 |
| 5 | CONCLUSION | 15 |

**Algorithm-**

1. INPUT THE NUMBER OF ROWS AND COLUMNS.

//M for rows & N for columns

2. DOMAIN IS THE SET OF ALL POSITIVE INTEGERS.

3. INPUT THE NUMBER OF SHADED POSITIONS & THEN THE REQUIRED POSITIONS.

4. BY CONSIDERING THE INPUT CONDITIONS, THE REQUIRED CHESS BOARD IS CONSTRUCTED AS FOLLOWS-

a) TRAVERSE THE ROWS AND COLUMNS OF THE BOARD.

b) IF THE REQUIRED PORTION THAT HAS TO BE SHADED IS REACHED THEN REPLACE IT BY '#'

c) TRAVERSE TILL THE LAST ELEMENT OF THE BOARD AND SHADE THE REQUIRED POSITIONS .

d) IF THE REQUIRED POSITION IS NOT PRESENT THEN REPEAT FROM STEP 3.

5. DISPLAY THE INPUT CHESS BOARD.

6. THEN FIND R1 VALUES, BY AGAIN TRAVERSING THE BOARD AND READING THE ELEMENTS THAT ARE NOT SHADED.

7. THEN FIND R2 VALUES AS FOLLOWS-

a) TRAVERSE THE BOARD AND ADD 2 MORE LOOPS TO COMPARE 1 ELEMENT PRESENT IN ROW WITH THE OTHER ELEMENT IN THE COLUMNN.

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

// for(j=(h+1);j<n;j++)

b) IF THE ELEMENT IN THE POSITION IS SHADED THEN JUST CONTINUE TRAVERSING.

c) IF THE ELEMENT THAT IS BEING COMPARED WITH SOME PARTICULAR ELEMENT IS NOT SHADED AS WELL AS NOT PRESENT IN THE SAME ROW OR COLUMN OF THAT ELEMENT THEN PAIR THOSE ELEMENTS FOR R2.

d) THEN CONTINUE STEPS 7 (a) (b) (c) UNTIL THE LAST ELEMENT IS REACHED.

8. THEN FIND R3 VALUES AS FOLLOWS-

a) TRAVERSE THE BOARD AND ADD 3 MORE LOOPS TO COMPARE 1 ELEMENT PRESENT IN ROW WITH 2 OTHER ELEMENTS IN THE COLUMNN.

// for(g=0;g<count;g++)

// for(h=g+1;h<count;h++)

// for(k=g+1;k<count;k++)

b) IF THE ELEMENT IN THE POSITION IS SHADED THEN JUST CONTINUE TRAVERSING.

c) IF THE ELEMENT THAT IS BEING COMPARED WITH SOME PARTICULAR ELEMENT IS NOT SHADED AS WELL AS NOT PRESENT IN THE SAME ROW OR COLUMN OF THAT ELEMENT THEN FORM TRIPLET OF THOSE ELEMENTS FOR R3.

d) THEN CONTINUE STEPS 8 (a) (b) (c) UNTIL THE LAST ELEMENT IS REACHED.

9. PRINT THE CORRESPONDING SET ELEMENTS OF R1,R2 & R3.

10. THEN FIND ROOKS POLYNOMIAL BY THE FORMULA 1+ R1x+ R2(x^2)+ R3(x^3).

11. PRINT THE ROOKS POLYNOMIAL AS THE OUTPUT.

12. STOP.

**Program-**

**Source code**

#include<stdio.h>

void main()

{

int a[5][5],m,n,s,g,h,q=1,max=0;

int b[10],k,flag=0,i,j,count=0,num,c[10],d[10];

printf("Problem: Design a program to construct the rook polynomial\n\n");

printf("Enter the number of rows and columns\n");

scanf("%d%d",&m,&n);

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

{

for(h=0;h<n;h++)

{

printf("%d\t",q);

a[g][h]=q;

q++;

}

printf("\n");

}

printf("Enter the number the shaded positions\n");

scanf("%d",&s);

printf("Enter the shaded positions\n");

for(k=0;k<s;k++)

{

scanf("%d",&b[k]);

}

printf("Now the given chess board is\n");

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

{

for(h=0;h<n;h++)

{

flag=0;

for(k=0;k<s;k++)

{

if(a[g][h]==b[k])

{

printf("#\t");

a[g][h]='$';

flag=1;

}

}

if(flag==0)

printf("%d\t",a[g][h]);

}

printf("\n");

}

printf("R1 values are: ");

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

{

for(h=0;h<n;h++)

{

if(a[g][h]!='$')

printf("(%d) ",a[g][h]);

}

}

k=0;

printf("\nR2 pairs are :\t");

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

{

for(h=0;h<n;h++)

{

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

{

for(j=(h+1);j<n;j++)

{

if(a[g][h]=='$')

continue;

else if(a[i][j]=='$')

continue;

else if(g!=i&&h!=j)

{

if((g\*n)+(h+1)<(i\*n)+(j+1))

{

c[k]=(g\*n)+(h+1);

d[k]=(i\*n)+(j+1);

}

else

{

d[k]=(g\*n)+(h+1);

c[k]=(i\*n)+(j+1);

}

printf(" (%d ,%d) ",c[k],d[k]);

k++;

count++;}

}

}

}

}

printf("\nR3 triplets are: ");

for(g=0;g<count;g++)

{

for(h=g+1;h<count;h++)

{

if(c[g]==c[h])

{

for(k=g+1;k<count;k++)

{

if(((d[g]==c[k])||(d[g]==d[k]))&&((d[h]==c[k])||(d[h]==d[k])))

{

max++;

printf(" ( %d , %d ,",c[g],d[g]);

if(d[g]==c[k])

printf(" %d ) ",d[k]);

else

printf("%d )",c[k]);

}

}

}

}

}

printf("\nRooks polynomial is= 1 + %dX ",((m\*n)-s));

if(count!=0)

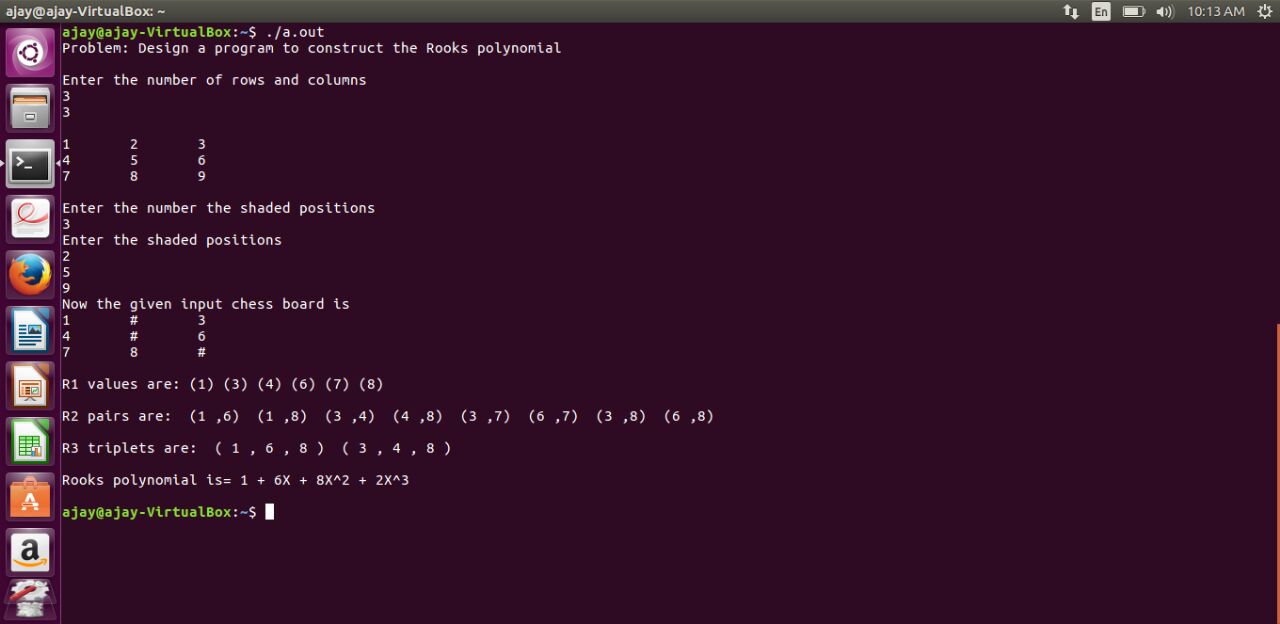
printf("+ %dX^2 ",count);

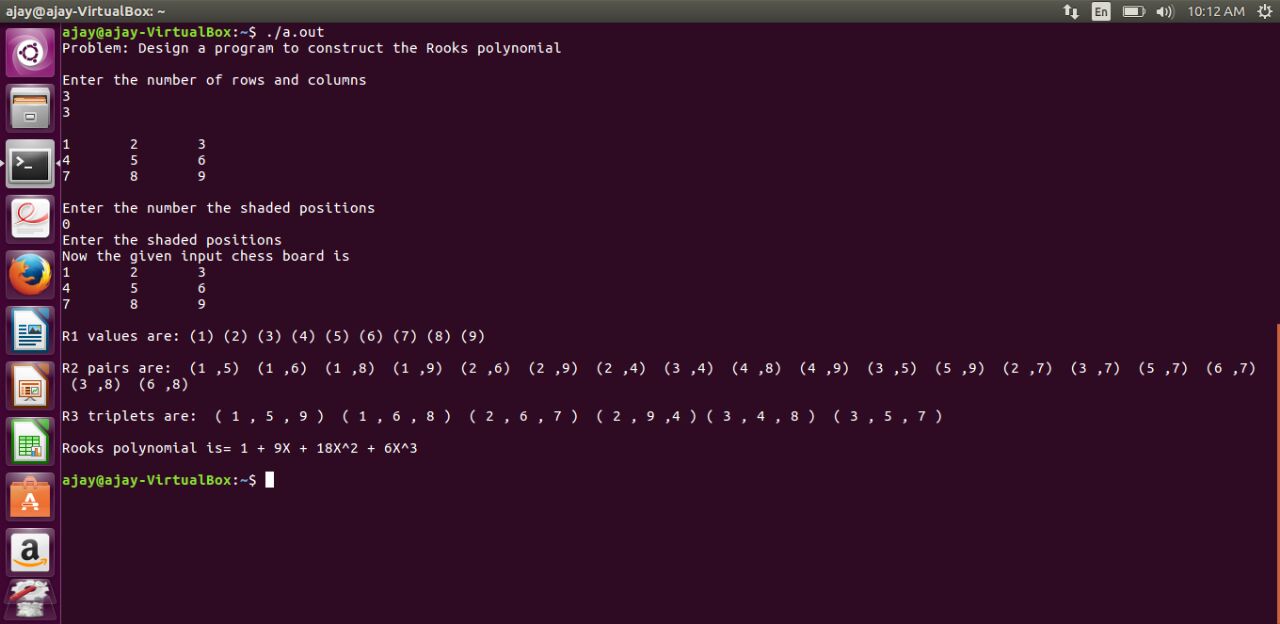
if(max!=0)

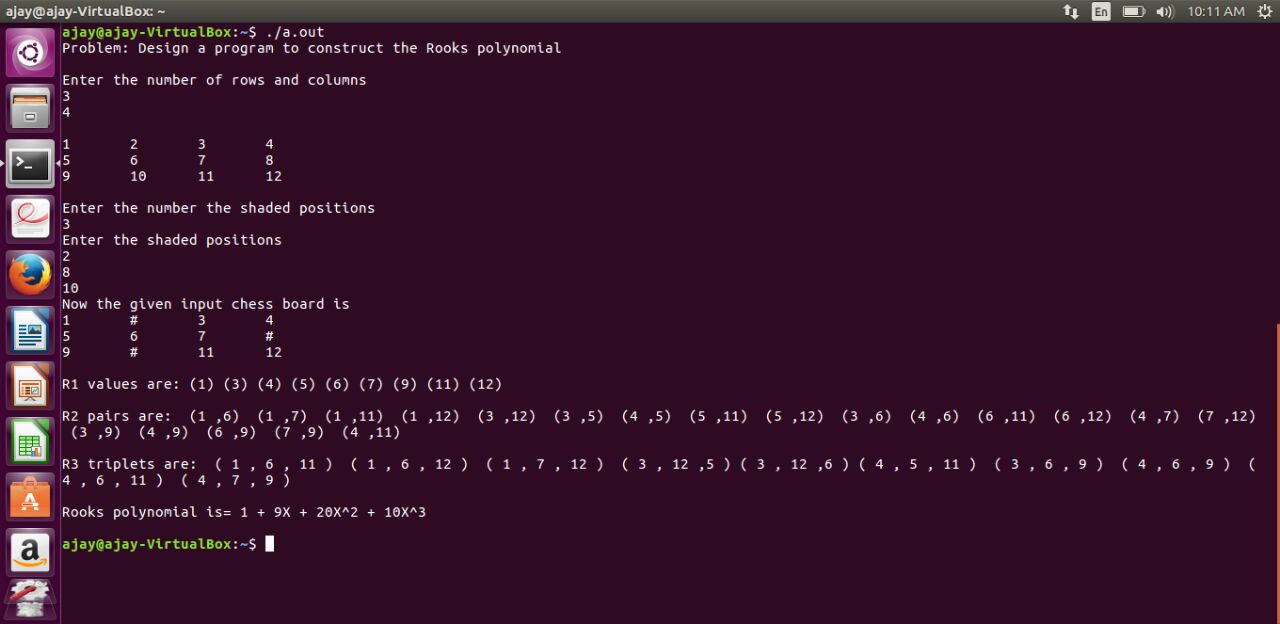
printf("+ %dX^3\n",max);

}

**Output-**







**Applications-**

1. In compiler design.
2. Hotspot prioritization system.
3. Digital Image processing.
4. In Embedded systems.
5. The rook polynomial corresponds to the complete bipartite graphs which are used extensively in online space, specifically in search advertising and e-commerce for similarity ranking.

**CONCLUSION-**

The Rook’s polynomial is obtained for any random board C by considering the number of ways to place non-attacking rooks on the checkerboard.