Finding fractions of n*n matrix:

```
#include<stdio.h>

#include<conio.h>

void dummy(float a){float *p=&a;}

void main()

{

float  a[10][10], b[10][10]; int nr,nc,r,c; clrscr();

printf("Enter no of rows and columns ");
scanf("%d %d",&nr,&nc);

printf("Enter %d numbers for 1st array \n", nr*nc);

for(r=0;r<nr;r++)for(c=0;c<nc;c++)scanf("%f",&a[r][c]);

printf("Enter %d numbers for 2nd array \n", nr*nc);

for(r=0;r<nr;r++)for(c=0;c<nc;c++)scanf("%f",&b[r][c]);

puts("Result elements are ");

for(r=0;r<nr;r++)

{

for(c=0;c<nc;c++)
```

```
{
printf("%10.2f",a[r][c]/b[r][c]);
}
printf("\n");
}
getch();
}
```

```
Enter no of rows and columns 2 2
Enter 4 numbers for 1st array
5.5 7.8 1.1 2.9
Enter 4 numbers for 2nd array
2.1 3.3 3.7 2.4
Result elements are
      2.62      2.36
      0.30      1.21
```

2.61

```
for( r=0;r<2;r++)
{
for(c=0;c<2;c++)
{
p("%10.2f",a[r][c]/b[r][c]);
}
p("\n");
}
```

a		b			
5.5 (0,0)	7.8 (0,1)	2.1 (0,0)	3.3 (0,1)	2.61	2.63
1.1 (1,0)	2.9 (1,1)	3.7 (1,0)	2.4 (1,1)	0.29	1.20

/ =

r c

0 0 1 2

1 0 1 2

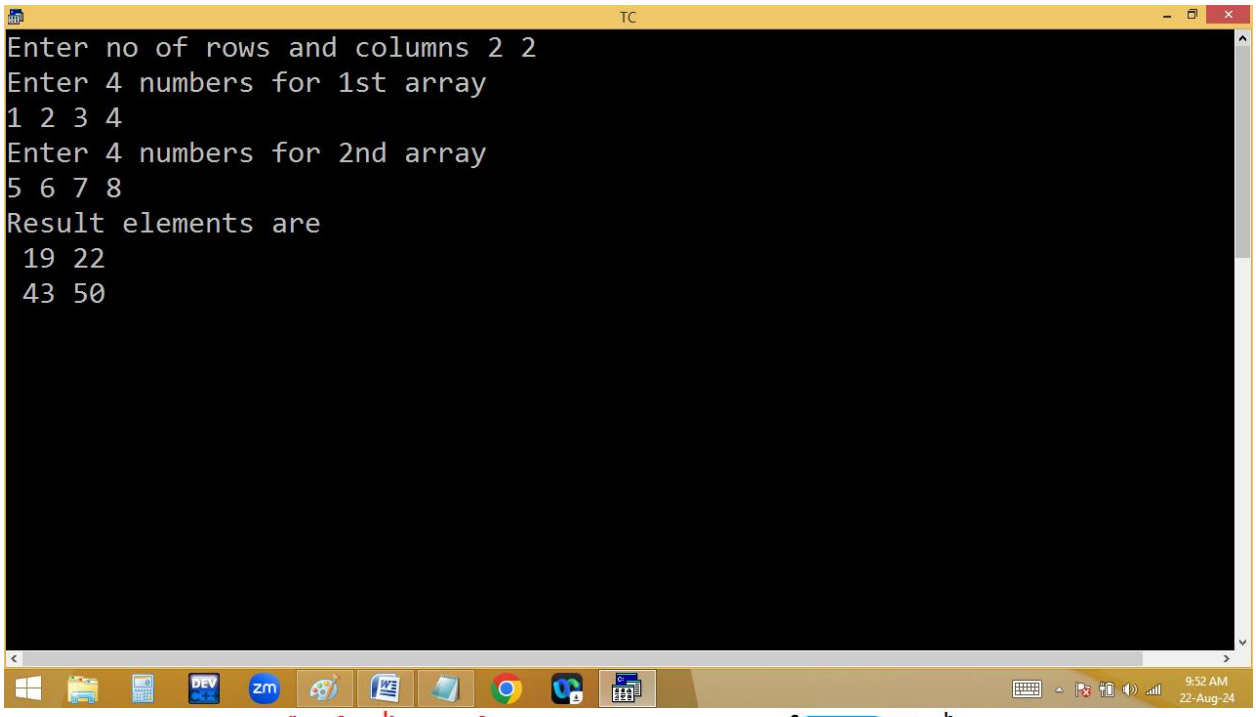
2

Matrix multiplication:

```
#include<stdio.h>

#include<conio.h>

void main()

{
```

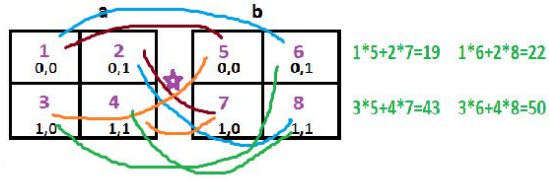
```
int  a[10][10], b[10][10], nr,nc,r,c,k,s; clrscr();
printf("Enter no of rows and columns ");
scanf("%d %d",&nr,&nc);
printf("Enter %d numbers for 1st array \n", nr*nc);
for(r=0;r<nr;r++)for(c=0;c<nc;c++)scanf("%d",&a[r][c]);
printf("Enter %d numbers for 2nd array \n", nr*nc);
for(r=0;r<nr;r++)for(c=0;c<nc;c++)scanf("%d",&b[r][c]);
puts("Result elements are ");
for(r=0;r<nr;r++)
{
for(c=0;c<nc;c++)
{
for(k=s=0;k<nc;k++)
{
s=s+a[r][k]*b[k][c];
}
printf("%3d",s);
```

```
}
printf("\n");
}
getch();
}
```




```
for(r=0;r<2;r++)
{
for(c=0;c<2;c++)
{
for(s=k=0;k<2;k++)
{
s+=a[r][k]*b[k][c];
}
p(s);
}
p("\n");
}
```

r	c	k	s
0	0	0 1 2	0+1*5=5+2*7=19 ✓
0	1	0 1 2	0+1*6=6+2*8=22 ✓
1	0	0 1 2	0+3*5=15+4*7=43 ✓
1	1	0 1 2	0+3*6=18+4*8=50 ✓



3-dimensional arrays:

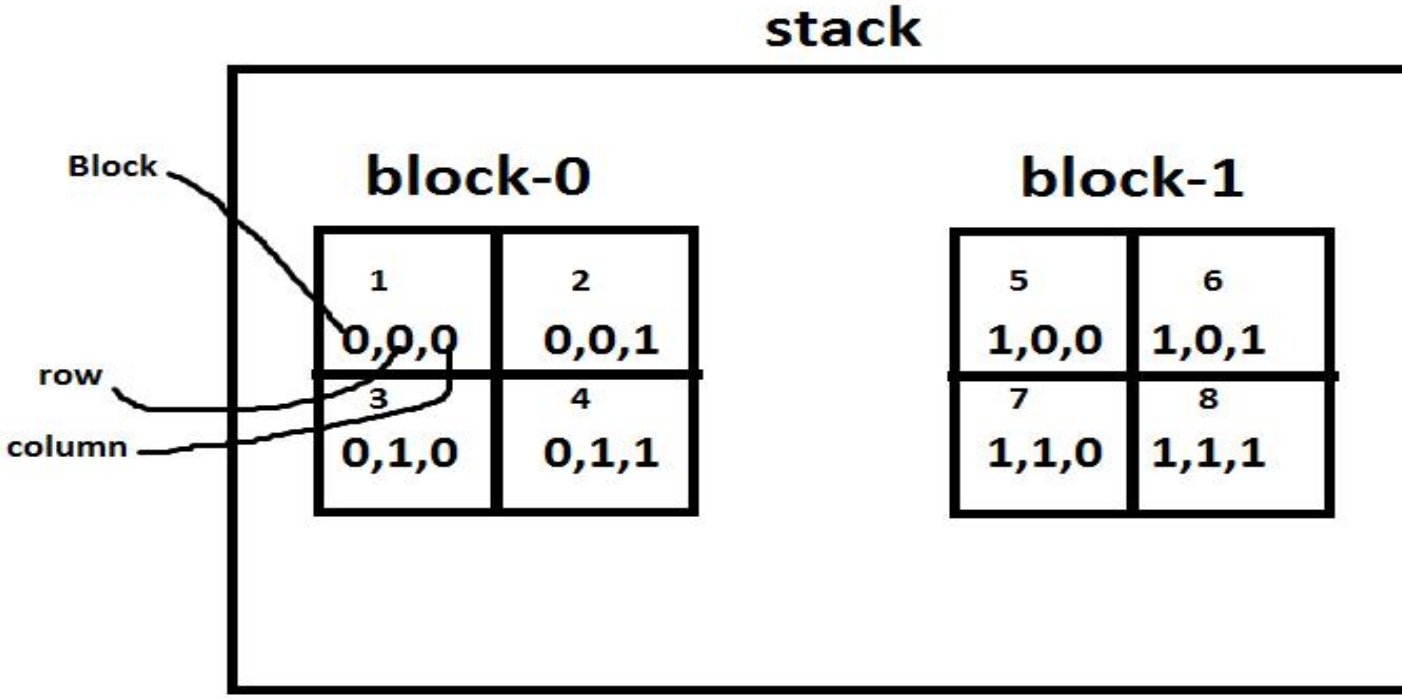
An array with several blocks, rows and columns.

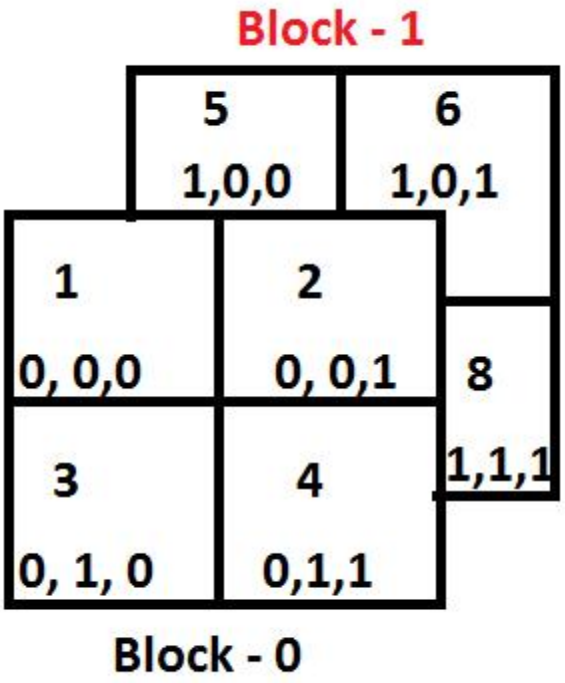
An array with 3 subscripting operators [ ] [ ] [ ].

Syntax:

datatype variable [blocks] [rows] [columns];

Eg: int a[2][2][2]={1,2,3,4,5,6,7,8};





Eg:

rows

int a[2][2][2]; an array with 3 subscripts [ ].

blocks columns

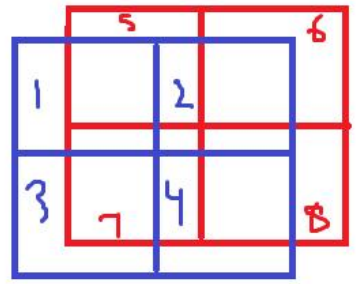
a[0][0][0]

b -0

1	2
3	4

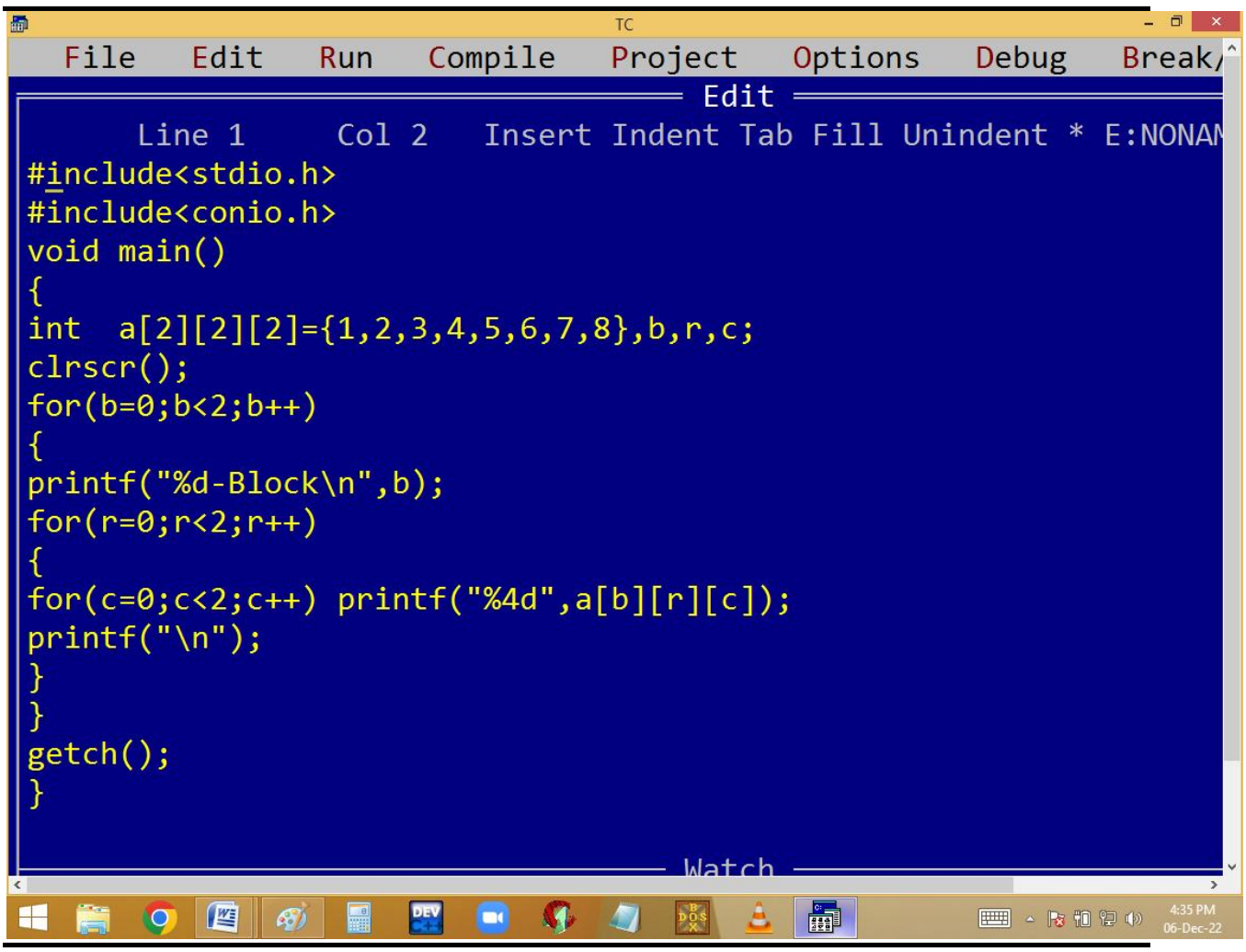
b-1

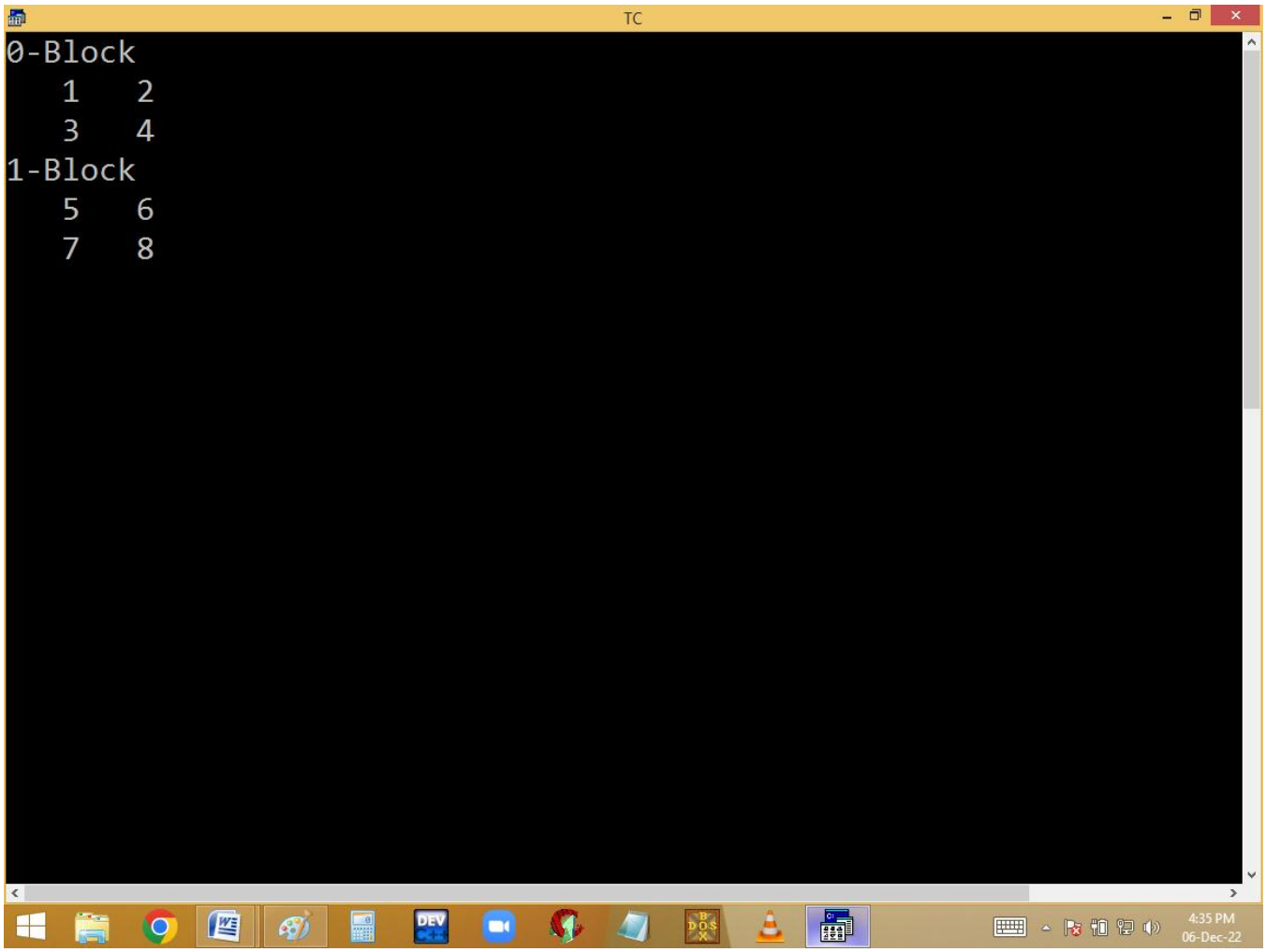
5	6
7	8



eg: int class[2][60][6];

datatype class[sections][stus][marks];





4-dimensional array:

An array with several sets, blocks, rows and columns.

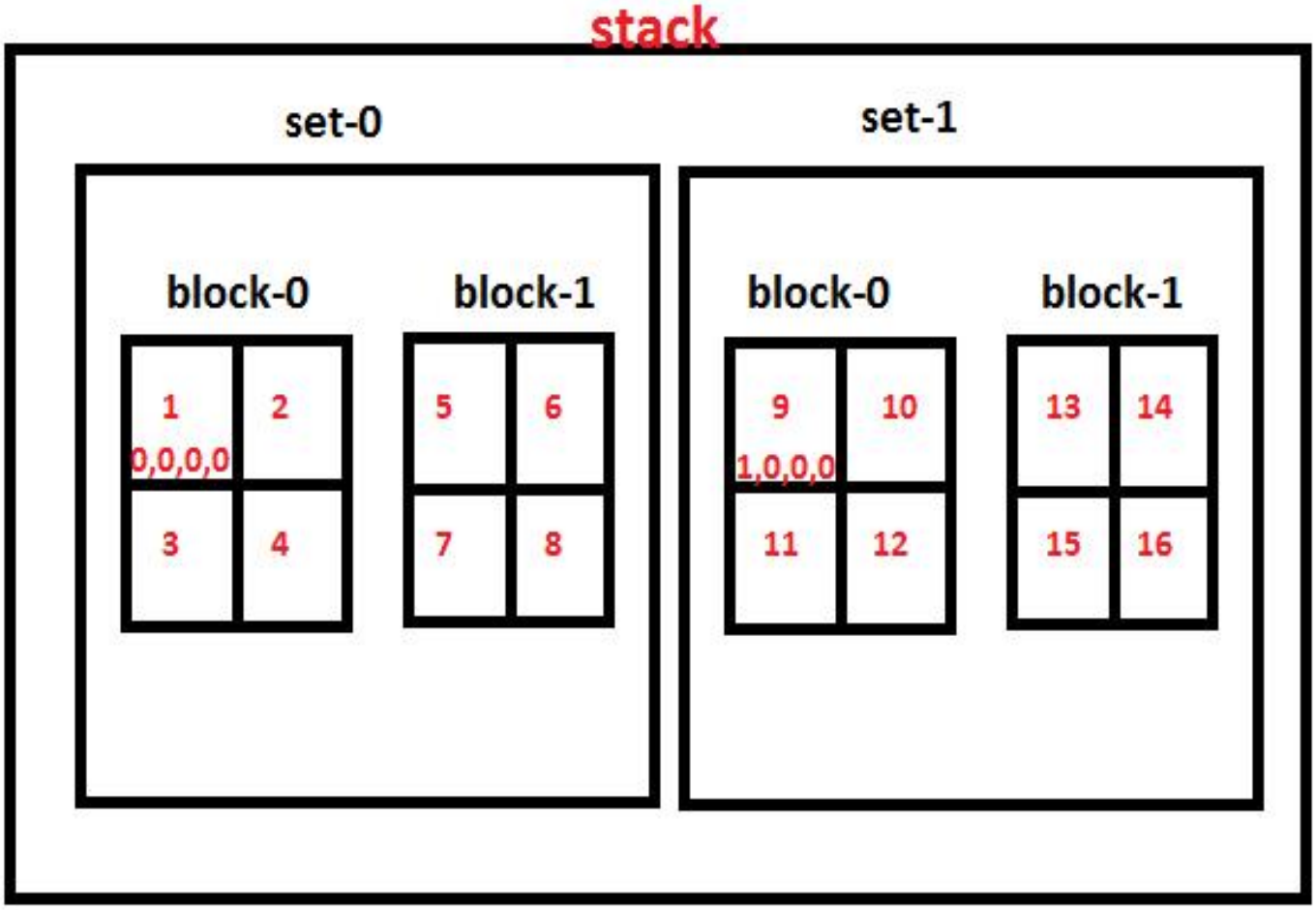
An array with 4 subscripting operators [ ] [ ] [ ] [ ] .

Syntax:

datatype variable [sets] [blocks] [rows] [cols];

eg:

```
int a[2] [2] [2] [2]= {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};
```



datatype var[class][sec][stu][marks];
int school[5][2][60][6];

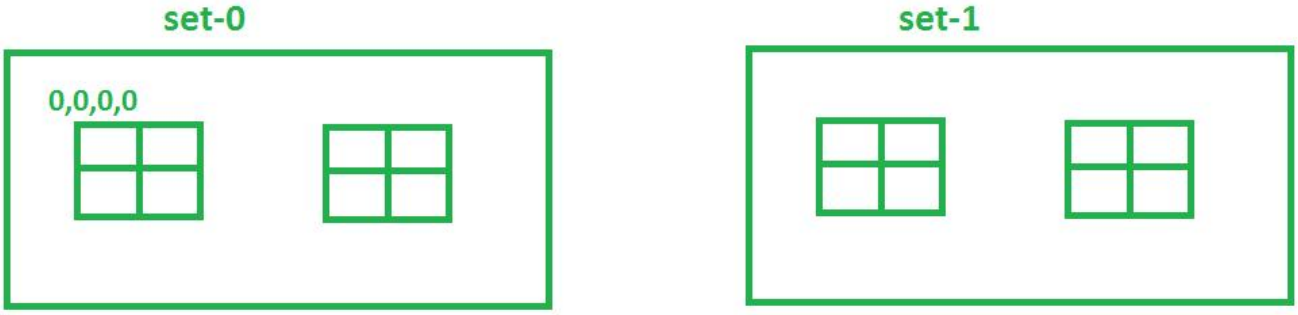
rows

set

int a[2][2][2][2]; an array with 3 subscripts [ ].

blocks

columns



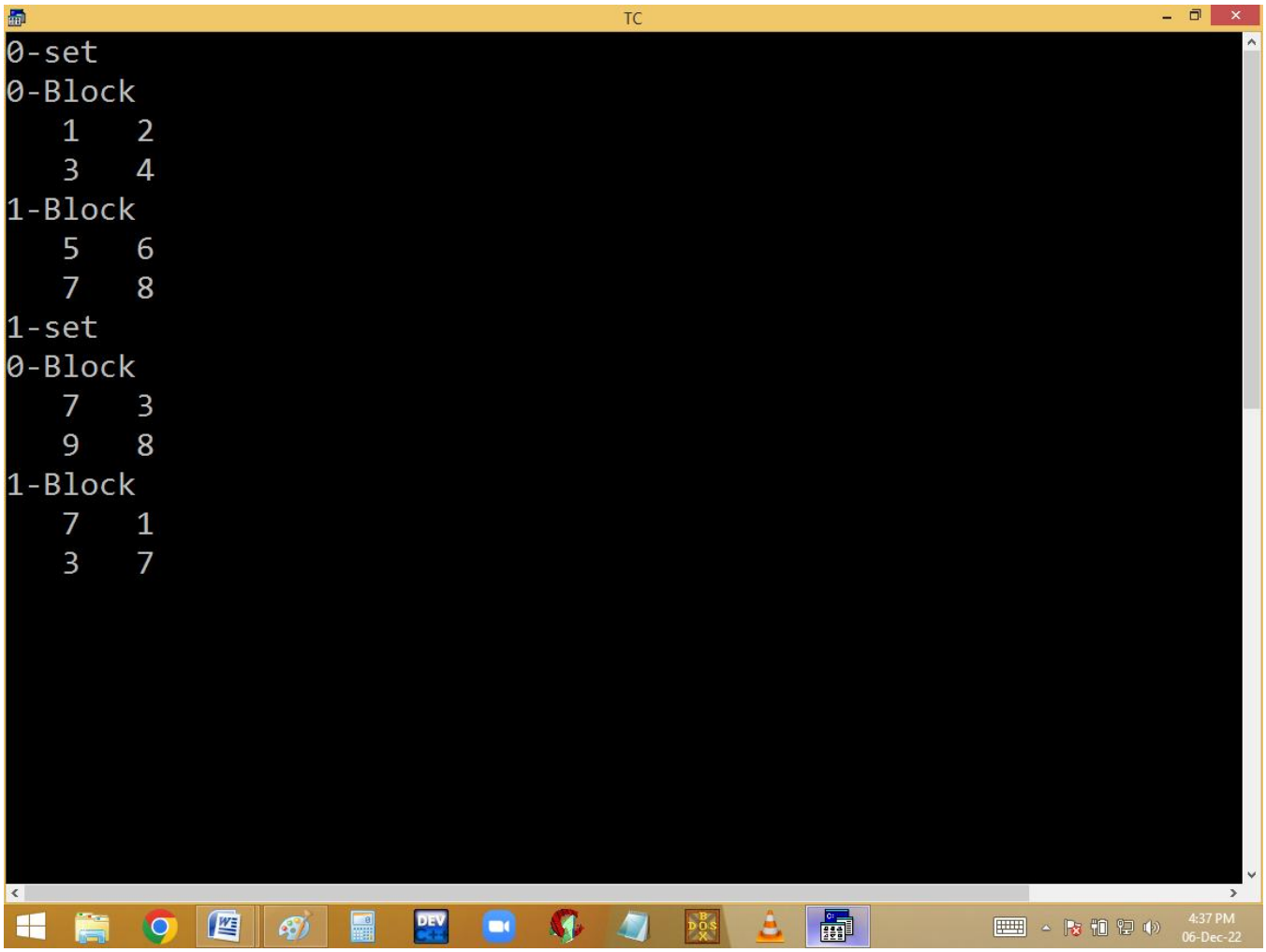
TC

File Edit Run Compile Project Options Debug Break/

Line 1 Col 47 Insert Indent Tab Fill Unindent * E:NONAME

```
#include<stdio.h>
#include<conio.h>
void main()
{
int  a[2][2][2][2]={1,2,3,4,5,6,7,8,7,3,9,8,7,1,3,7},s,b,r,c;
clrscr();
for(s=0;s<2;s++)
{
printf("%d-set\n",s);
for(b=0;b<2;b++)
{
printf("%d-Block\n",b);
for(r=0;r<2;r++)
{
for(c=0;c<2;c++) printf("%4d",a[s][b][r][c]); printf("\n");
}
}
}
getch();
}
```

4:37 PM
06-Dec-22



STRINGS

A group of characters is called string.

It is alpha-numeric. i.e. in a string we can store both alphabets, numbers and special char.

It is one dimensional character array.

It is an implicit pointer.

It is a derived data type.

It is a non primitive data type.

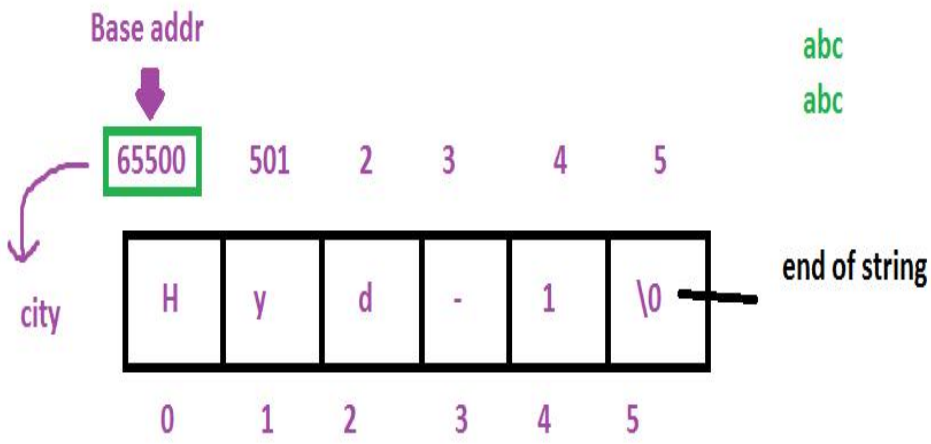
Syntax: char var[size]=”string”;

Eg:

char city[6]=”Hyd-1”;

char city[ ]=”Hyd-1”;

1. One byte for null char.
2. Str var size never smaller than string. Otherwise error.
3. We can’t copy a string with = operator. We have to use strcpy().
4. We can’t compare two strings with == op. we have to use strcmp ./ char by char



abc
abc

int sub[6]; int array
char name[40]="Sri";

city2 = 65550

if(city==city2)

city = city2
65500 = 65550