

Arrays in c

CHAPTER 31



SURESH TECHS

C PROGRAMMING COURSE

B Tech College



CR

- Let's say you are the **CR** of the class(**B.Tech 1st year cse**) and your class teacher asked you to list **12th class percentages** of all the **60 students**
- Write a **c program** to note down **12th class percentages of 60 students** and ask them to attend **special classes on Sunday** if their **percentage is less than 70.**

```
#include<stdio.h>
int main() {
    float roll1P,roll2P,roll3P,roll4P,roll5P;
    printf("Roll no 1, what is your percentage? ");
    scanf("%f",&roll1P);
    printf("Roll no 2, what is your percentage? ");
    scanf("%f",&roll2P);
    printf("Roll no 3, what is your percentage? ");
    scanf("%f",&roll3P);
    printf("Roll no 4, what is your percentage? ");
    scanf("%f",&roll4P);
    printf("Roll no 5, what is your percentage? ");
    scanf("%f",&roll5P);

    //<70 - need to attend special classes on sunday
    if(roll1P<70){
        printf("Roll no 1, please attend special class\n");
    }
    if(roll2P<70){
        printf("Roll no 2, please attend special class\n");
    }
    if(roll3P<70){
        printf("Roll no 3, please attend special class\n");
    }
    if(roll4P<70){
        printf("Roll no 4, please attend special class\n");
    }
    if(roll5P<70){
        printf("Roll no 5, please attend special class\n");
    }
    return 0;
}
```


What happens

- float percentages[5];

percentages[0]=20

percentages[3]=90

0	20
0	
0	
0	90
0	
Array	

Indexes

0

1

2

3

4

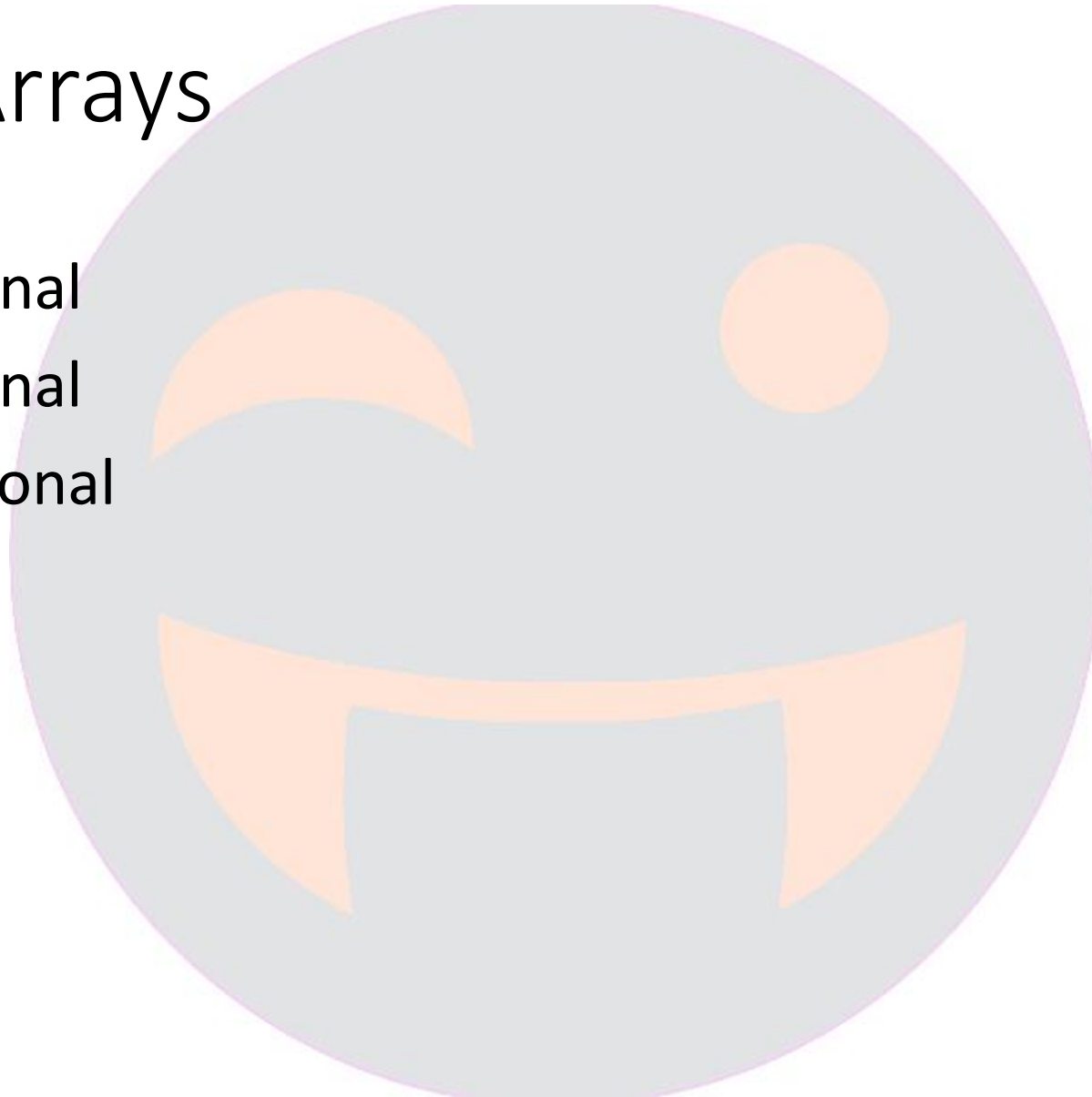
What is an array?

- An **array** is a collection of one or more values of the same data-type stored in **contiguous memory** locations.
- The **data-type** can be **user-defined** or even any other **primitive data-type**.
- Elements of an array can be accessed with the **same array name** by specifying the **index number** as the location in memory.

```
float value =percentages[5];  
percentages[3]=76;
```

Types of Arrays

- One-dimensional
- Two-dimensional
- Multi-dimensional



One Dimensional Array

14	75	59	37	75	63
0	1	2	3	4	5

- While declaring a one-dimensional array in C, the **data type can be of any type**, and also, we can give any name to the array, just like naming a variable.
- Syntax:
 - `datatype arrayname[size];`
- Ex: `float students[60];`

Initializing an array

- In an **uninitialized array**, all the elements initially contain **garbage values**
- If we want to initialize all elements of integer array to zero
- `datatype arrayname[size] = {0};`

Size of the array

- `sizeof(arrayname)/sizeof(datatype)`
- `float students[60];`
- `sizeof(students)/sizeof(float)`
- `Sizeof(students)/sizeof(students[0])`

```
float students[60]={56,77,45.3,98.3};  
int size = sizeof(students)/sizeof(students[0]);  
for(int i=0;i<size;i++){  
    printf("%.2f\n",students[i]);  
}
```

Initializing an array

- Explicitly initialize them at their declaration.

```
float students[60]={56,77,45.3,98.3};  
int size = sizeof(students)/sizeof(students[0]);  
for(int i=0;i<size;i++){  
    printf("%.2f\n",students[i]);  
}
```

We can **skip writing size** of the array within square brackets if we initialize array elements explicitly within the list at the time of declaration. In that case, it will pick elements list size as array size.

Accessing array elements

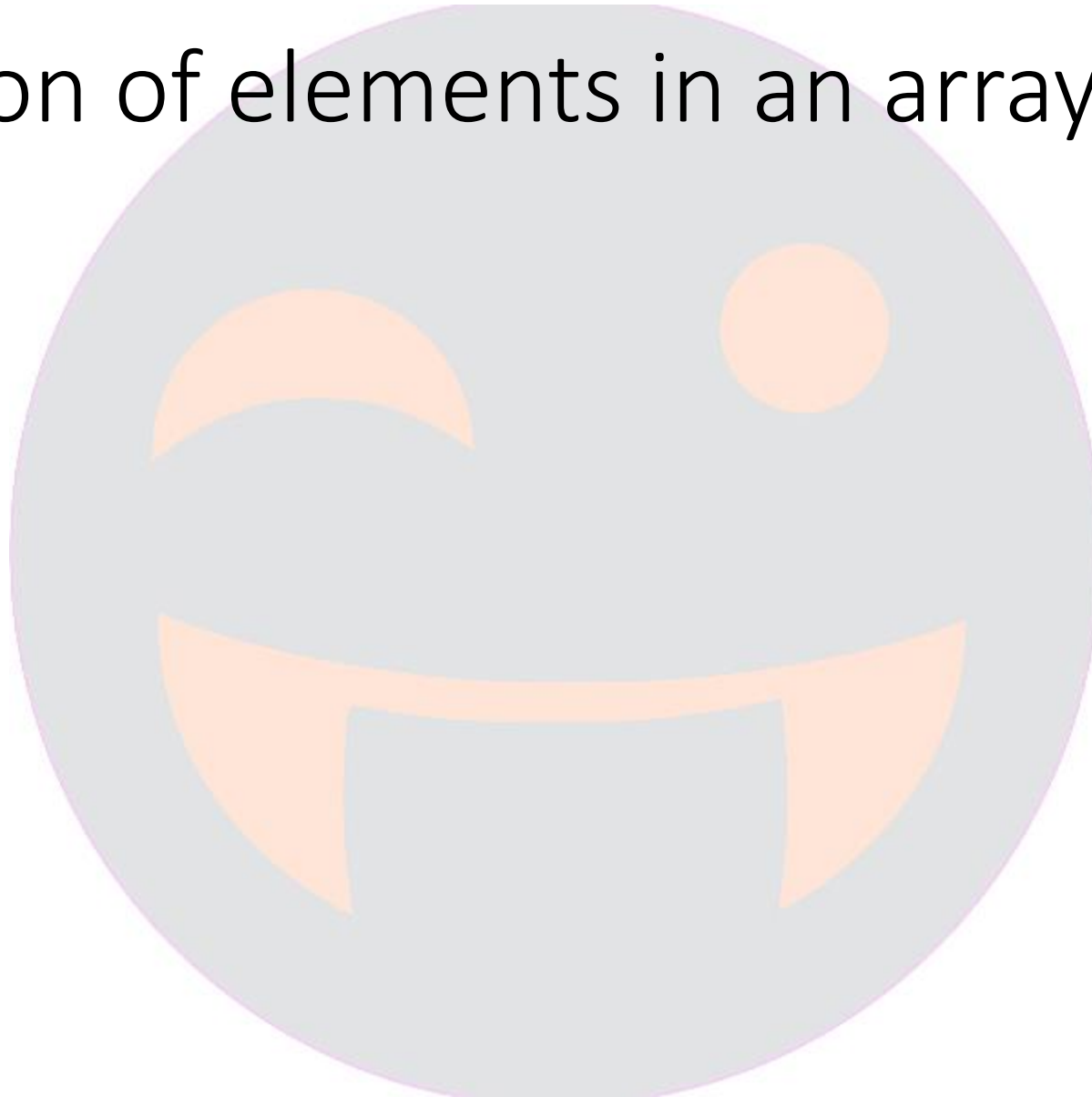
- Elements are accessed by specifying the **array name and the index value within the square brackets**.
- `arrayname[index];`
- Array indexing **starts from 0 and ends with size-1**.
- If we try to access array elements **out of the range**, the compiler will not show any error message; rather, it will return some **garbage value**.

Rules for declaring one-dimensional array

- Before using and accessing, we must declare the array variable.
- In array, indexing starts from 0 and ends at size-1. For example, if we have `arr[10]` of size 10, then indexing of elements ranges from 0 to 9.
- We must include data-type and variable name while declaring one-dimensional arrays in C.
- We can initialize them explicitly when the declaration specifies array size within square brackets is not necessary.
- Each element of the array is stored at a contiguous memory location with a unique index number for accessing.

Initialization of elements in an array

- **Compile time**
- **Run time**



Run time

- Runtime initialization is also known as **dynamic-initialization**. Array elements are initialized at the runtime after successfully compiling the program.
- `scanf("%f", &students[0]);`

Compile time

- **Compile-Time initialization** is also known as **static-initialization**. In this, array elements are initialized when we declare the array implicitly.
- `<data_type> <array_name> [array_size]={list of elements};`

```
float students[60]={50,60,70,80,76,55,89};  
int size = sizeof(students)/sizeof(students[0]);  
for(int i=0;i<size;i++){  
    printf("%.2f\n",students[i]);  
}
```

Let us modify our program to take percentages dynamically

```
#include<stdio.h>
int main() {
    float students[60];
    int size = sizeof(students)/sizeof(students[0]);
    for(int i=0;i<size;i++){
        printf("Roll no %d, what is your percentage? ",i+1);
        scanf("%f",&students[i]);
    }
    printf("====A Few moments later====\n\n");
    for(int i=0;i<size;i++){
        if(students[i]<70){
            printf("Roll no %d, please attend special class\n",i+1);
        }
    }
    return 0;
}
```

What if someone tells invalid percentage(naughty people like me:D)

```
#include<stdio.h>
int main(){
    float students[5];
    int size = sizeof(students)/sizeof(students[0]);
    for(int i=0;i<size;i++){
        printf("Roll no %d, what is your percentage? ",i+1);
        scanf("%f",&students[i]);
        if(students[i]<0 || students[i]>100){
            printf("Over action cheyaku, malli cheppu");
            i=i-1;
        }
    }
    printf("====A Few moments later====\n\n");
    for(int i=0;i<size;i++){
        if(students[i]<70){
            printf("Roll no %d, please attend special class\n",i+1);
        }
    }
    return 0;
}
```


If it is repeated 5 times then don't take percentages

```
#include<stdio.h>
int main(){
    float students[5];
    int size = sizeof(students)/sizeof(students[0]);
    int count=0;
    for(int i=0;i<size;i++){
        printf("Roll no %d, what is your percentage? ",i+1);
        scanf("%f",&students[i]);
        if(students[i]<0 || students[i]>100){
            count++;
            if(count==5){
                printf("me chavu meeru chavandi...");
                exit(0);
            }
            printf("Over action cheyaku, malli cheppu");
            i=i-1;
        }
    }
    printf("====A Few moments later====\n\n");
    for(int i=0;i<size;i++){
        if(students[i]<70){
            printf("Roll no %d, please attend special class\n",i+1);
        }
    }
    return 0;
}
```

Task: Average percentage of entire class

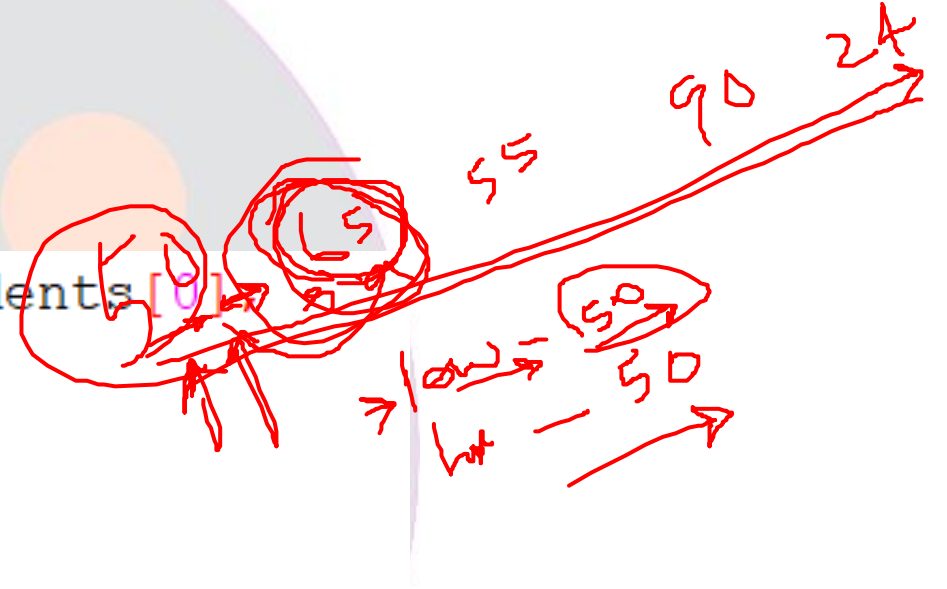
```
float totalSum = 0;
for(int i=0;i<size;i++){
    totalSum = totalSum+students[i];
}
printf("Average percentage: %f",totalSum/size);
```

Advantages

- **Code Optimization:** Less code to access the data.
- **Ease of traversing:** By using the for loop, we can retrieve the elements of an array easily.
- **Ease of sorting:** To sort the elements of the array, we need a few lines of code only.
- **Random Access:** We can access any element randomly using the array.

Task: What is the highest percentage and lowest percentage?

```
float lowest=students[0],highest=students[0];
for(int i=1;i<size;i++){
    if(students[i]<lowest){
        lowest=students[i];
    }
    if(students[i]>highest){
        highest=students[i];
    }
}
printf("lowest: %f, highest: %f",lowest,highest);
```



Copying into another array

```
#include<stdio.h>
int main() {
    int marks=76;
    int myMarks;
    myMarks = marks;
    printf("My marks: %d",myMarks);
    return 0;
}
```

```
#include<stdio.h>
int main() {
    int marks[5]={50,20,60,55,76};
    int myMarks[5];
    myMarks = marks;
    return 0;
}
```

Not possible

Solution to copy one array into another

```
#include<stdio.h>
int main() {
    int marks[5]={50,20,60,55,76};
    int myMarks[5];
    //    myMarks = marks;
    for(int i=0;i<5;i++) {
        myMarks[i]=marks[i];
    }
    for(int i=0;i<5;i++) {
        printf("%d\t",myMarks[i]);
    }
    return 0;
}
```

Sort an array

```
for(int i=0;i<5;i++){  
    for(int j=i+1;j<5;j++){  
        if(marks[i]>marks[j]){  
            int temp = marks[i];  
            marks[i]=marks[j];  
            marks[j]=temp;  
        }  
    }  
}
```



Array

int sem1[]

14	75	59	37	75	63
0	1	2	3	4	5

int sem2[]

74	43	39	19	42	55
0	1	2	3	4	5

int marks[]

14	75	59	37	75	63
74	43	39	19	42	55

**But what if you want to represent
two or more arrays at a time?**

Two rows, six columns

int marks[2][6]

**We can represent 12
items at a time**

All semester marks

Two dimensional array 🤗 🤗

Represented as rows and columns

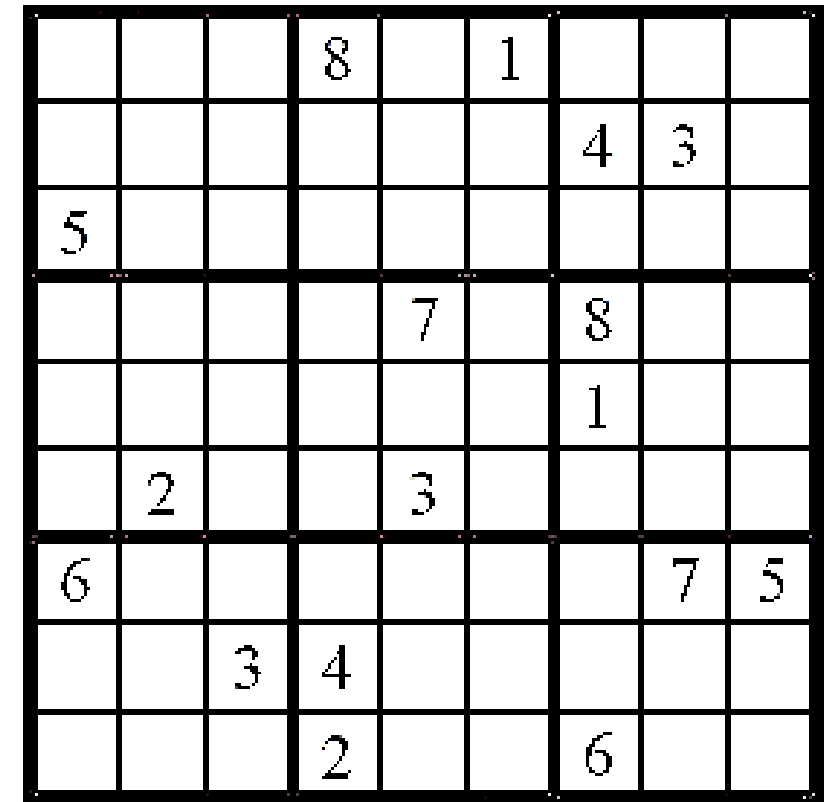
14	75	59	37	75	63
74	43	39	79	42	55
44	75	59	28	75	53
54	43	99	82	42	98
77	75	82	31	75	63
39	43	87	65	42	76
84	75	59	37	75	63
74	43	39	79	42	55

`int marks[8][6]`

We can represent 48 items at a time

Two dimensional array

- Two dimensional array is represented as **collection of rows and columns**
- Array of arrays



			8		1			
						4	3	
5								
				7		8		
						1		
	2			3				
6							7	5
		3	4					
			2			6		

Declaration

- `data_type array_name[rows][columns];`
- `int sudoku[9][9];`
- `int marks[8][6];` //8 semesters, 6 subject marks

Represented as rows(8) and columns(6)

Semesters

Subjects(M, E, C, P, T, H)

Col 0 Col 1 Col 2 Col 3 Col 4 Col 5

Row 0	14	75	59	37	75	63
Row 1	74	43	39	19	42	55
Row 2	44	75	59	28	75	53
Row 3	54	43	99	82	42	98
Row 4	77	75	82	31	75	63
Row 5	39	43	87	65	42	76
Row 6	84	75	59	37	75	63
Row 7	74	43	39	79	42	55

int marks[8][6]

What is your **Physics** marks in **3rd** semester?

```
printf("%d",marks[2][3]);
```

After writing **supplementary exams** you passed **first semester maths, second semester physics**. So please update the marks

```
marks[0][0]=56;
```

```
marks[1][3]=67;
```

Declaring 2d array

- `int marks[8][6]={14,75,59,37,...42,5};` //all the 48 marks
- `int marks[8][6]={{14,75,59,37,75,63},{74,43,39,19,42,55},{}}...{}}`
- Dynamically taking values using `scanf`

Let us write a program to take Ruju marks

```
#include<stdio.h>
int main() {
    int rajuMarks[8][6]={
        {14,75,59,37,75,63},
        {74,43,39,19,42,55},
        {44,75,59,28,75,53},
        {54,43,99,82,42,98},
        {77,75,82,31,75,63},
        {39,43,87,65,42,76},
        {84,75,59,37,75,63},
        {74,43,39,79,42,55},
    };

    printf("%d\n",rajuMarks[0][0]);
    printf("%d\n",rajuMarks[0][1]);
    printf("%d\n",rajuMarks[0][2]);
    printf("%d\n",rajuMarks[1][0]);
    printf("%d\n",rajuMarks[3][6]);
    return 0;
}
```



14
75
59
74
77

Print all the marks of all the semesters of Raju

```
#include<stdio.h>
int main(){
    int rajuMarks[8][6]={
        {14,75,59,37,75,63},
        {74,43,39,19,42,55},
        {44,75,59,28,75,53},
        {54,43,99,82,42,98},
        {77,75,82,31,75,63},
        {39,43,87,65,42,76},
        {84,75,59,37,75,63},
        {74,43,39,79,42,55},
    };

    for(int i=0;i<8;i++){
        for(int j=0;j<6;j++){
            printf("%d\t",rajuMarks[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Let us dynamically take Ramu marks

```
#include<stdio.h>
int main(){
    int rajuMarks[8][6]={
        {14,75,59,37,75,63},
        {74,43,39,19,42,55},
        {44,75,59,28,75,53},
        {54,43,99,82,42,98},
        {77,75,82,31,75,63},
        {39,43,87,65,42,76},
        {84,75,59,37,75,63},
        {74,43,39,79,42,55},
    };

    int ramuMarks[8][6];
    printf("Enter Ramu marks\n");
    for(int i=0;i<8;i++){
        for(int j=0;j<6;j++){
            scanf("%d",&ramuMarks[i][j]);
        }
    }
}
```

```
printf("Raju marks\n");
for(int i=0;i<8;i++){
    for(int j=0;j<6;j++){
        printf("%d\t",rajuMarks[i][j]);
    }
    printf("\n");
}

printf("Ramu marks\n");

for(int i=0;i<8;i++){
    for(int j=0;j<6;j++){
        printf("%d\t",ramuMarks[i][j]);
    }
    printf("\n");
}

return 0;
}
```

Add marks of Raju and Ramu

```
printf("Combined marks of raju and ramu\n");  
for(int i=0;i<8;i++){  
    for(int j=0;j<6;j++){  
        printf("%d\t",rajuMarks[i][j]+ramuMarks[i][j]);  
    }  
    printf("\n");  
}
```

Let us take a simple example(3x3)

```
#include<stdio.h>
int main(){
    int rajuMarks[3][3];
    int ramuMarks[3][3];
    printf("Enter Raju, Ramu marks seperated by space\n");
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            scanf("%d %d",&rajuMarks[i][j],&ramuMarks[i][j]);
        }
    }

    printf("Raju marks\n");
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            printf("%d\t",rajuMarks[i][j]);
        }
        printf("\n");
    }

    printf("Ramu marks\n");

    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            printf("%d\t",ramuMarks[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```


Perform addition and multiplication of
matrices



Transpose the matrix

	0	1	
0	20	10	30
1	40	33	45
2	80	45	87

2 → 2st
3 → 3rd
5 → 5th

	0	1	2
0	20		
1	10		
2			

arr[2][1] = arr[1][2]

20	40	80
10	33	45
30	45	87

Let us represent 60 students marks

```
#include<stdio.h>
int main() {
    int student1Marks[8][6];
    int student2Marks[8][6];
    int student3Marks[8][6];
    int student4Marks[8][6];
    int student5Marks[8][6];
    int student6Marks[8][6];
    int student7Marks[8][6];
    int student8Marks[8][6];
    return 0;
}
```

**We need to write 60 two dimensional arrays
for 60 students**

Student1

14	75	59	37	75	63
74	43	39	19	42	55
44	75	59	28	75	53
54	43	99	82	42	98
77	75	82	31	75	63
39	43	87	65	42	76
84	75	59	37	75	63
74	43	39	79	42	55

Student2

14	75	59	37	75	63
74	43	39	19	42	55
44	75	59	28	75	53
54	43	99	82	42	98
77	75	82	31	75	63
39	43	87	65	42	76
84	75	59	37	75	63
74	43	39	79	42	55

Student3

14	75	59	37	75	63
74	43	39	19	42	55
44	75	59	28	75	53
54	43	99	82	42	98
77	75	82	31	75	63
39	43	87	65	42	76
84	75	59	37	75	63
74	43	39	79	42	55

Student4

14	75	59	37	75	63
74	43	39	19	42	55
44	75	59	28	75	53
54	43	99	82	42	98
77	75	82	31	75	63
39	43	87	65	42	76
84	75	59	37	75	63
74	43	39	79	42	55

Student5

14	75	59	37	75	63
74	43	39	19	42	55
44	75	59	28	75	53
54	43	99	82	42	98
77	75	82	31	75	63
39	43	87	65	42	76
84	75	59	37	75	63
74	43	39	79	42	55

Multi dimensional arrays

14	75	59	37	75	63
74	43	39	19	42	55
44	75	59	28	75	53
54	43	99	82	42	98
77	75	82	31	75	63
39	43	87	65	42	76
84	75	59	37	75	63
74	43	39	79	42	55

.....

14	75	59	37	75	63
74	43	39	19	42	55
44	75	59	28	75	53
54	43	99	82	42	98
77	75	82	31	75	63
39	43	87	65	42	76
84	75	59	37	75	63
74	43	39	79	42	55

For one student: studentMarks[8][6]

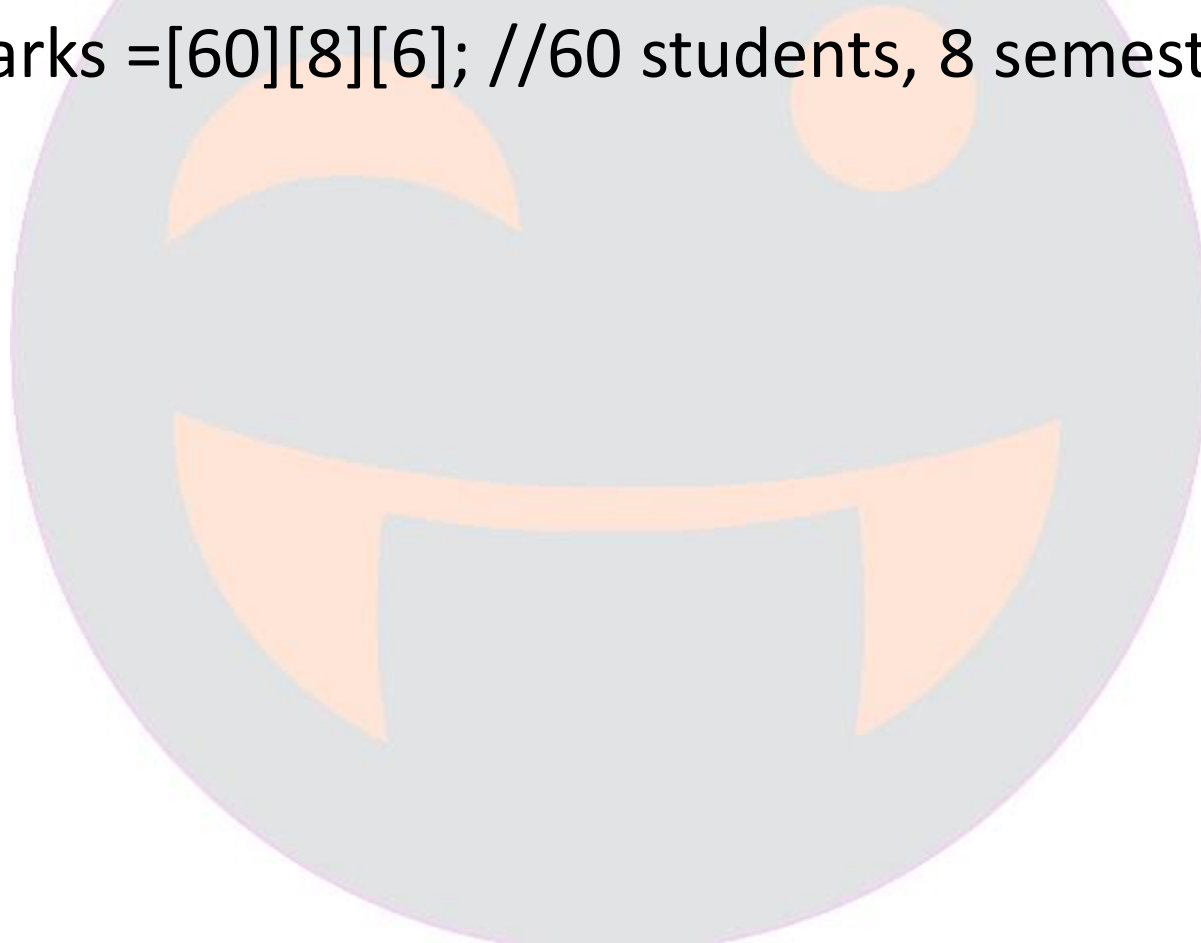
For 60 student: studentMarks[60][8][6]

Three dimensional array

0	1	2	3	4	5	59
---	---	---	---	---	---	---	---	---	---	---	---	----

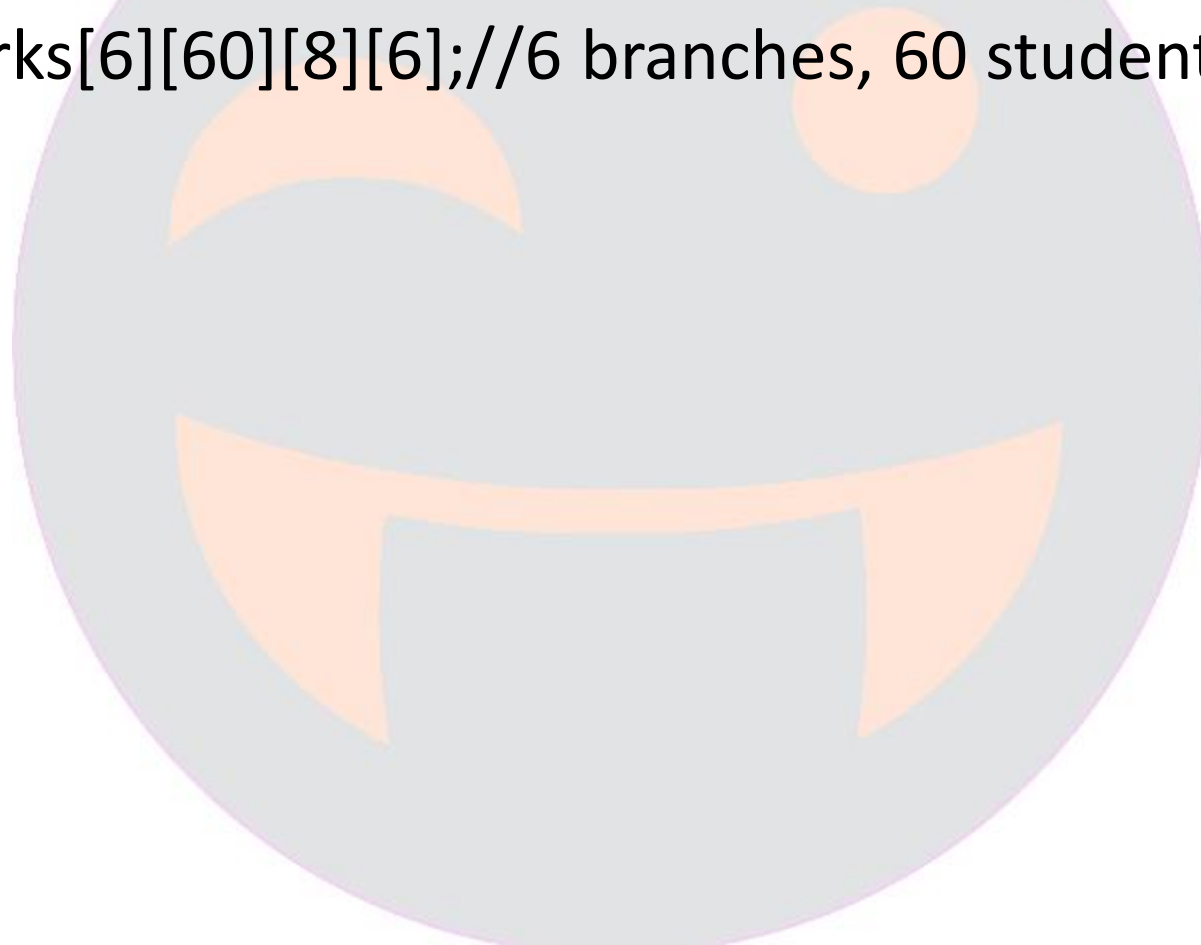
Let us write a program to take marks of 60 students

- `int cseClassMarks =[60][8][6]; //60 students, 8 semesters, 8 subjects`



Write a program to take marks of all the branches(cse,it,ece,eee,mech,civil)

- `int collegeMarks[6][60][8][6];`//6 branches, 60 student, 8 semesters,6 subjects



Write a program to take marks of all the colleges under JNTU-K(cse,it,ece,eee,mech,civil)

- `int allCollegeMarks[740][6][60][8][6];`//6 branches, 60 student, 8 semesters,6 subjects

What next?

- Strings in detail

