Paper 102:
Programming & Problem solving through C

Lecture-15:Unit-II
Pointers & Multidimensional Arrays

```
//dynamic 1-d array
#include<stdio.h>
void main()
{
        int *n,size,i;
        clrscr();
        printf("\nHow many elements:");
        scanf("%d",&size);
        n=(int *)malloc(size * sizeof(int));
        for(i=0;i<size;i++)
        {printf("\n enter a number:");
        scanf("%d",(n+i));
        }
        printf("\n the output is:\n");
        for(i=0;i<size;i++)
        {
                printf("%d\t",*(n+i));
        }
        getch();
}
```

```
//A pointer to an array
#include<stdio.h>
#define MAXC 10
void inputdata(int (*n)[MAXC],int r, int c);
void main()
{
        int (*a)[MAXC];
        int r,c,i,j;
        clrscr();
        printf("\n how many rows?");
        scanf("%d",&r);
        printf("\n how many cols?");
        scanf("%d",&c);
        a=(int *) malloc(r * sizeof(int));
        printf("\n input data");
        inputdata(a,r,c);

        for(i=0;i<r;i++)
        {
          for(j=0;j<c;j++)
          {
            printf("%d\t",*(*(a+i) + j));
           }
           printf("\n");
        }
        getch();
}
```

```
void inputdata(int (*n)[MAXC],int r,int c)
{
  int i,j;
  for(i=0;i<r;i++)
  {
      for(j=0;j<c;j++)
      {
            printf("\nenter number:");
            scanf("%d",(*(n+i)+j));
                  }
            }
}
```

```
//an array of pointers
#include<stdio.h>
#define MAXR 10
void inputdata(int *n[MAXR],int r, int c);
void main()
{
        int *b[MAXR];
        int r,c,i,j;
        clrscr();
        printf("\n how many rows?");
        scanf("%d",&r);
        printf("\n how many cols?");
        scanf("%d",&c);
        for(i=0;i<r;i++)
                b[i]=(int *) malloc(c * sizeof(int));
        printf("\n input data");
        inputdata(b,r,c);

        printf("\n b array\n");
        for(i=0;i<r;i++)
        {
                for(j=0;j<c;j++)
                {
                        printf("%d\t",*(*(b+i) + j));
                }
                printf("\n");
        }
        getch();
}
```

```
void inputdata(int *n[MAXR],int r,int c)
{
   int i,j;
   for(i=0;i<r;i++)
   {
     for(j=0;j<c;j++)
     {
           printf("\nenter number:");
           scanf("%d",(*(n+i)+j));
     }
    }
}
```

Creating a 2-d array Dynamically

```
#include<stdio.h>
void main()
{
        int **array1;
        int nrows,ncols,i,j;
        clrscr();
        printf("\n enter the rows:");
        scanf("%d",&nrows);
        printf("\n enter the cols:");
        scanf("%d",&ncols);
 //  Allocate an array of pointers.
 //  Then initialize each pointer to a dynamically
 //  allocated row.
 array1 = (int **) malloc( nrows * sizeof(int*));
 if( array1 == NULL){
        printf("Out of memory");
 }
 for(i = 0; i < nrows; i++){
        array1[i] = (int * )malloc(ncols * sizeof(int));
 }
```

```
for (i=0; i<nrows; i++){
        for (j=0;j<ncols;j++)
  {             printf("\n enter a
number:");
                scanf("%d",&array1[i][j]);
        }
  }
  for (i=0; i<nrows; i++){
        for (j=0;j<ncols;j++)
                printf("%8d",array1[i][j]);
                printf("\n");

        }
  }
```

//using command line arguments

```
#include<stdio.h>
#include<string.h>
void main(int argc,char *argv[])
{
        int i,j;
        if(argc==1)
                exit();
        printf("number of arguments=%d",argc);
        for(i=0;i<argc;i++)
        {       puts(argv[i]);
                printf("%d",strlen(argv[i]));

        }


}
```

1. **argc is the total number of arguments**
2. ***argv[] is an array of strings, storing the list of arguments including the program name**
3. **To run create an exe file and execute from the dos prompt:**

   D:\>cmdline *argument1 argument2*