```
// Array insert and delete

#include <stdio.h>
int main()
{
   int a[100], n, i, data, pos;
   printf("\nEnter the no of inputs");
   scanf("%d",&n);

   for(i=0;i<n;i++)                          // input of an array
   {
       printf("\nenter the value for a[%d]",i);
       scanf("%d",&a[i]);
   }
//______
   printf("\nArray looks like below...");     // output of an array
   for(i=0;i<n;i++)
      printf("%5d\t", a[i]);

//______
    // Now we will insert some element to an array        PORE MUKHE
BOLCHI EKHN CODE TA JUST DEKHTE THAK OK ??
    // Inserting an element into an array

    //scanning an element
    printf("\nEnter the data to be insert... ");
    scanf("%d",&data);

    //scanning for position
    do{
      printf("\nEnter the position (for insertion) [range 0 to %d] ...
",n-1);
      scanf("%d",&pos);
    }while(pos<0 || pos>n);


    // 0 1 2 3 4 5 6 7    // positions
    // 1 2 3 9 7 4 5 6       array size 7 n=7    pos=11 ?? NO   pos<n
pos ???
    i=n;   // size of an array
    while(i>pos)                       // SHIFTER
    {
        a[i]=a[i-1];
        i--;
    }
    a[pos] = data;     // insert the element into that position
    n++;     // no of data in that array will be increased ....

//______
   printf("\nAfter Insertion, Array looks like below...");     // output
of an array
   for(i=0;i<n;i++)
      printf("%5d\t", a[i]);
```

```
//_____________________ARRAY DELETION
    // 0 1 2 3 4 5 6 7    // positions
    // 1 2 3 9 7 4 5 6    array size 8 n=8    pos 3

    //scanning for position
    do{
      printf("\nEnter the position (for deletion) [range 0 to %d] ...
",n-1);
      scanf("%d",&pos);
    }while(pos<0 || pos>n);     //pos=11 ?? NO   pos<n   pos ???

    data=a[pos];
    printf("\nThis data %5d is deleted now...", data);

    i=pos;
    while(i<n-1)         //left shifting
    {
       a[i]= a[i+1];
       i++;
    }
    n--;     // no of data in that array will be decreased ....
//______
   printf("\nAfter Deletion, Array looks like below...");     // output
of an array
   for(i=0;i<n;i++)
      printf("%5d\t", a[i]);

   return 0;
}
```

```
//matrix addition (r1 = r2  and c1= c2)

#include <stdio.h>
int main()
{                              //matrix (2D array) declaration with
initialization
    int mat1[10][10]={{12,34,45,24},{56,67,78,39},{42,46,70,30}}, m1r=3,
m1c=4, i, j;
    int mat2[10][10]={{12,34,45,24},{56,67,78,39},{42,46,70,30}}, m2r=3,
m2c=4, mat3[10][10];

    // matrix data scanning (input)
//    printf("\nEnter row\t");     scanf("%d", &m1r);
//    printf("\nEnter column\t");  scanf("%d", &m1c);

/*    printf("\nEnter Data Values...\n");
    for(i=0;i<m1r;i++)
       for(j=0;j<m1c;j++)
          scanf("%5d", &mat1[i][j]);
*/
    printf("\nMatrix Looks like...\n");
    if((m1r==m2r) && (m1c==m2c))   //matrix addition conditon approved
here
    {

        for(i=0;i<m1r;i++)
        {
           for(j=0;j<m1c;j++)
               mat3[i][j]=mat1[i][j] + mat2[i][j];
        }
     }

    printf("\nMatrix Looks like...\n");
    for(i=0;i<m1r;i++)
    {
        for(j=0;j<m1c;j++)
          printf("%5d", mat3[i][j]);
        printf("\n");
    }
   return 0;
}
```

```
//matrix Subtraction (r1 = r2  and c1= c2)

#include <stdio.h>
int main()
{                              //matrix (2D array) declaration with
initialization
    int mat1[10][10]={{12,34,45,24},{56,67,78,39},{42,46,70,30}}, m1r=3,
m1c=4, i, j;
    int mat2[10][10]={{12,34,45,24},{56,67,78,39},{42,46,70,30}}, m2r=3,
m2c=4, mat3[10][10];

    // matrix data scanning (input)
//    printf("\nEnter row\t");     scanf("%d", &m1r);
//    printf("\nEnter column\t");  scanf("%d", &m1c);

/*    printf("\nEnter Data Values...\n");
    for(i=0;i<m1r;i++)
       for(j=0;j<m1c;j++)
          scanf("%5d", &mat1[i][j]);
*/
    printf("\nMatrix Looks like...\n");
    if((m1r==m2r) && (m1c==m2c))   //matrix subtraction conditon approved
here
    {

        for(i=0;i<m1r;i++)
        {
           for(j=0;j<m1c;j++)
               mat3[i][j]=mat1[i][j] - mat2[i][j];
        }
     }

    printf("\nMatrix Looks like...\n");
    for(i=0;i<m1r;i++)
    {
        for(j=0;j<m1c;j++)
          printf("%5d", mat3[i][j]);
        printf("\n");
    }
   return 0;
}
```

```
// Matrix Multiplication (r1 = r2  and c1= c2)

#include <stdio.h>
int main()
{                              //matrix (2D array) declaration with
initialization
    int mat1[10][10]={{1,3,4,4},{5,2, 4,3},{2,6,0,0}}, m1r=3, m1c=4, i,
j, k;
    int mat2[10][10]={{1,4}, {4,2},{5,6}, {7,3}}, m2r=4, m2c=2,
mat3[10][10];

    // matrix data scanning (input)
//    printf("\nEnter row\t");     scanf("%d", &m1r);
//    printf("\nEnter column\t");  scanf("%d", &m1c);

/*    printf("\nEnter Data Values...\n");
    for(i=0;i<m1r;i++)
       for(j=0;j<m1c;j++)
          scanf("%5d", &mat1[i][j]);
*/

//  formula   r1*c1   r2*c2      =>   if c1==r2 then resultant matrix
will looks like r1 *c2

    printf("\nMatrix Looks like...\n");
    if((m1c==m2r))   // Condition of matrix multiplication (approved
here)
    {
        for(i=0;i<m1r;i++)       // first mat row
        {
           for(j=0;j<m2c;j++)        //second mat column
           {
               mat3[i][j]=0;                                          //s =0;
               for(k=0; k<m2r; k++)
               {
                   mat3[i][j]= mat3[i][j] + mat1[i][k]*mat2[k][j];
               }
           }
        }
     }

    printf("\nMatrix Looks like...\n");
    for(i=0;i<m1r;i++)
    {
        for(j=0;j<m2c;j++)
          printf("%5d", mat3[i][j]);
        printf("\n");
    }
   return 0;
}
```

```
// String
// Define: String is a character array ended with a NULL character ('\0')

//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#include<stdio.h>
int main()
{
 char a[10]={'a','y','a','n'};   // Q: is it a string or not ??
 int n=4;                         // A: chracter array but not a string

 for(i=0;i<n;i++)
   printf("%c", a[i]);
   return 0;
}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#include<stdio.h>
int main()
{
 char a[10]={'a','y','a','n','\0'};   // Q: is it a string or not ??
 int n=4;                         // A: Yes this is a string

 for(i=0;i<n;i++)
   printf("%c", a[i]);
   return 0;
}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#include<stdio.h>
int main()
{
 char a[10]="ayan";        // A: Yes this is a string
 i=0;
 while(a[i] != '\0')
 {
     printf("%c", a[i]);
     i++;
 }
   return 0;
}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ String with pointer
#include<stdio.h>
int main()
{
 char a[10]="ayan", *p;
 p=a;           // array name means address of its first cell

 while(*p != '\0')
 {
     printf("%c", *p);
     p++;
 }
   return 0;
}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ String with pointer
#include<stdio.h>
```

```
int main()
{
 char a[10]="ayan", *p;
 p=a;           // array name means address of its first cell
 printf("%s",*p);    // eliminate the loops
   return 0;
}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Problem with %s
format
#include<stdio.h>
int main()
{
 char a[30];
 printf("Enter your name plz");
 scanf("%s", a);              // i/p:    Ayan Kumar Dey   // 14
characters     //a =  &a[0]
                              //  a will store   "Ayan"

 printf("Hi, %s, How are you",a);    // Expected o/p: Hi Ayan Kumar Dey,
How are you
                                     // Actual O/P:   Hi Ayan, How are
you
   return 0;
}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Solution For %s
format
#include<stdio.h>
int main()
{
 char a[30];
 printf("Enter your name plz");
 scanf("%[^\n]s", a);              // %s = scan any characters untill
space or '\n'
                                 // %[^\n]s = scan any characters untill
'\n'
 printf("Hi, %s, How are you",a);    // Expected o/p: Hi Ayan Kumar Dey,
How are you
   return 0;
}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Solution For %s
format (2nd approach)
#include<stdio.h>
int main()
{
 char a[30];
 printf("Enter your name plz");
 gets(a);
 puts("Hi,"); puts(a); puts(", How are you");
   return 0;
}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Play with Pointers
and Strings
#include<stdio.h>
int main()
```

```
{
    char str1[]="ayan", str2[50];
    char *s= "Hello", *t;

    // array is very responsible to its values
    // pointer is irresponsible

    str2=str1;  // semantical error
    str1=str2; // semantical error
    s=t;           // correct statment
    t=s;
    s=str1;     //correct
    str1=s;   // error


    str2="mithun da";  // Error //array name =  address of a[0] = &a[0]
    t="Nachun Na";     // Correct
}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Exercise of String
"Let Us C"
#include<stdio.h>
int main()
{
 printf(5+"Good Morning");   // 5 characters will be skipped from the
base address | Output: Morning
 printf("%c\n", "abcdefgh"[4]);      // 4th character considering 0 at
base index  | output 'e'
 return 0;
}

#include<stdio.h>
int main()
{
 char str1[]={'a','y','a','n','0'};
 char str2[]="ayan";
 printf("\n%s", str1);    //  output:   ayana
124443$%5h36y88%!@dt3472r31221u49326t46r27321
 printf("\n%s", str2);    //  output:   ayan
 return 0;
}

#include<stdio.h>
int main()
{
 char str1[]="ayan";
 char str2[]="ayan";

 if(str1==str2)      // values are same but address are diffrent
   puts("Bujhte perechis ?");
 else
   puts("kichui bujhis ni");             // output:  kichui bujhis ni
 return 0;
}
```

```c
#include<stdio.h>
int main()
{
 char str1[2]="A";
 printf("\n%c", str1[0]);  // output:  A
 printf("\n%c", str1[1]);  // output:  NULL
 printf("\n%s", str1);     // output:  A
 return 0;
}
```