

1. 1D array Criminal suspect:

```
#include<stdio.h>
#include <stdbool.h>

int main(){
    int size = 10;
    bool match = true;
    //declaring suspect array
    float suspect[size];
    //declaring criminal array
    float criminal[size];
    //read 10 input values into suspect array from keyboard
    printf("Enter the 10 chromosomes of the suspect separated by
spaces: \n");
    for (int i = 0; i < size; i++)
    {
        scanf(" %f", &suspect[i]);
    }
    //read 10 input values into criminal array from keyboard
    printf("Enter the 10 chromosomes of the criminal separated
by spaces: \n");
    for (int i = 0; i < size; i++)
    {
        scanf(" %f", &criminal[i]);
    }

    for(int i=0; i<size;i++)
    {
        if(suspect[i]!=criminal[i])
```

```

    {
        match=false;
    }
}
if(match==true)
{
    printf("The two profile match! \n");
}
else
{
    printf("The two profile doesnot match");
}
return 0;
}

```

Output:

```

} ; if ($?) { .\matchingProfileA }
Enter the 10 chromosomes of the suspect separated by spaces:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Enter the 10 chromosomes of the criminal separated by spaces:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
The two profile match!
PS C:\TBC\Sem2\POP_Project\Weekly Assignments\Practical4> cd "c:\
m2\POP_Project\Weekly Assignments\Practical4\" ; if ($?) { gcc ma
ProfileA.c -o matchingProfileA } ; if ($?) { .\matchingProfileA }
Enter the 10 chromosomes of the suspect separated by spaces:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5
Enter the 10 chromosomes of the criminal separated by spaces:
1.3 0.3 9.5 8.7 5.8 4.1 3.2 2.3 6.2 6.9
The two profile doesnot match
PS C:\TBC\Sem2\POP_Project\Weekly Assignments\Practical4> 

```

2. Matching Profile B

```
3. #include <stdio.h>
4. #include <stdbool.h>
5.
6. int main()
7. {
8.     int s_sizeR=1,sizeR=3, sizeC=10,match_count=0;
9.     //Multiple criminal (10 chromosomes)
10.    float criminals[sizeR][sizeC];
11.    float suspects[s_sizeR][sizeC];
12.    bool match;
13.
14.    //read multiple profile of 10 values
15.    for(int i=0;i<sizeR;i++)
16.    {
17.        printf("Enter the 10 chromosomes of the %d
criminal: \n",i+1);
18.        for(int j=0; j<sizeC;j++)
19.        {
20.            scanf("%f",&criminals[i][j]);
21.        }
22.        printf("\n");
23.    }
24.
25.    for(int i=0;i<s_sizeR;i++)
26.    {
27.        printf("Enter the 10 chromosomes of the %d
suspect: \n",i+1);
28.        for(int j=0; j<sizeC;j++)
29.        {
30.            scanf("%f",&suspects[i][j]);
```

```

31.         }
32.         printf("\n");
33.     }
34.
35.
36.     for(int sus=0;sus<s_sizeR;sus++)
37.     {
38.         //for matching chromosome of each suspect with
        each criminal
39.         //one suspect 3 criminal
40.         for(int cri=0;cri<sizeR;cri++)
41.         {
42.             match = true;// to check whether the
            chromosome matches or not
43.             printf("Checking if suspect %d chromosome
            matches with %d criminal: \n",sus+1,cri+1);
44.             for(int chromo=0;chromo<sizeC;chromo++)
45.             {
46.                 //checkinng if the suspects chromosome
                matches with criminal
47.                 if(criminals[cri][chromo]!=suspects[sus][chr
                omo])
48.                 {
49.                     //if matches returns false if any one
                    chromosome didn't matched
50.                     match = false;
51.                 }
52.             }
53.             if(match== false)
54.             {

```

```
55.         printf("sus %d profile doesnot match
    \n",sus+1);
56.     }
57.     else
58.     {
59.         printf("sus %d profile matches with %d
    criminal!!!\n",sus+1,cri+1);
60.         //counting total dna matched if returned true
61.         match_count+=1;
62.         //breaks if match found
63.     }
64. }
65.     printf("\n");
66.
67. }
68.     printf("Suspect chromosome matches with %d
    criminal profile",match_count);
69.
70.     return 0;
71. }
```

Output:

```
atchingProfiles }
Enter the 10 chromosomes of the 1 criminal:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

Enter the 10 chromosomes of the 2 criminal:
1.3 0.3 9.5 8.7 5.8 4.1 3.2 2.3 6.2 6.9

Enter the 10 chromosomes of the 3 criminal:
6.3 9.3 4.3 6.4 7.5 2.9 3.0 4.1 5.3 6.5

Enter the 10 chromosomes of the 1 suspect:
2.3 3.3 4.5 6.7 7.8 2.1 3.2 4.3 5.2 6.5

Checking if suspect 1 chromosome matches with 1 criminal:
sus 1 profile matches with 1 criminal!!!
Checking if suspect 1 chromosome matches with 2 criminal:
sus 1 profile doesnot match
Checking if suspect 1 chromosome matches with 3 criminal:
sus 1 profile doesnot match

Suspect chromosome matches with 1 criminal profile
PS C:\TBC\Sem2\POP_Project\Weekly Assignments\Practical4> █
```

3. Repeated offender A

```
/*  
Pseudocode:  
1. Begin  
2. Define file pointer, and assign address  
3. Check if the file is NULL or not  
4. Assign Variables  
5. Read and take input of suspect dna and criminal from file  
6. check if DNA matches or not with file input (listed criminal  
dna)  
7. If matches count the number of time it's matched  
8. Print output.  
9. Close file.  
10. End  
*/
```

```
#include <stdio.h>  
#include <stdbool.h>
```

```
int main()  
{  
    FILE * fp;  
    fp = fopen("dna_input.txt", "r");  
  
    if (fp == NULL)  
    {  
        printf("File doesnot exist!!!");  
        return 1;  
    }  
}
```

```
int cri_sizeR=5,sizec=10, total_match=0;
float suspect[sizec];
float criminal[cri_sizeR][sizec];
bool match;

//taking input from file of suspect dna
for(int i=0; i<sizec;i++)
{
    fscanf(fp,"%f",&suspect[i]);
}

//taking input from file of criminal dna
//printf("Reading criminal dna: ");
for(int i=0;i<cri_sizeR;i++)
{
    for(int j=0;j<sizec;j++)
    {
        fscanf(fp,"%f",&criminal[i][j]);
    }
}

//checking matching dna
for(int cri=0; cri<cri_sizeR;cri++)
{
    match= true;
    for(int i=0; i<sizec;i++)//runs 10 times
    {
        if(criminal[cri][i]!=suspect[i])
        {
            //if anyone of the dna didn't matched it returns false
            match=false;
        }
    }
}
```



```

    }
}
//if matched adds total matched dna
if(match==true)
{
    printf("The suspect dna matches with criminal
%d.\n",cri);
    //counts the total matched dna
    total_match+=1;
}
/*else
{
    printf("The suspect dna doesnot matches with criminal
%d.\n",cri);
}*/

}
printf("Suspect is a repeated offender.\n");
printf("\nSuspect dna matches with total %d
criminals.\n",total_match);
fclose(fp);
return 0;
}

```

edOffenderA.c M × dna_input.txt × matchingProfile

Weekly Assignments > Practical4 > dna_input.txt

| | | | | | | | | | | |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | 2.3 | 3.3 | 4.5 | 6.7 | 7.8 | 2.1 | 3.2 | 4.3 | 5.2 | 6.5 |
| 2 | 2.3 | 3.3 | 4.5 | 6.7 | 7.8 | 2.1 | 3.2 | 4.3 | 5.2 | 6.5 |
| 3 | 1.3 | 0.3 | 9.5 | 8.7 | 5.8 | 4.1 | 3.2 | 2.3 | 6.2 | 6.9 |
| 4 | 6.3 | 9.3 | 4.3 | 6.4 | 7.5 | 2.9 | 3.0 | 4.1 | 5.3 | 6.5 |
| 5 | 6.1 | 9.4 | 4.5 | 6.6 | 7.4 | 2.8 | 3.2 | 4.4 | 5.0 | 6.0 |
| 6 | 2.3 | 3.3 | 4.5 | 6.7 | 7.8 | 2.1 | 3.2 | 4.3 | 5.2 | 6.5 |

Output:

```

• PS C:\TBC\Sem2\POP_Project\Weekly Assignments\Practical4> cd ..\Sem2\POP_Project\Weekly Assignments\Practical4\" ; if ($?) { go
OffenderA.c -o repeatedOffenderA } ; if ($?) { .\repeatedOffe
The suspect dna matches with criminal 0.
The suspect dna matches with criminal 4.
Suspect is a repeated offender.

Suspect dna matches with total 2 criminals.
• PS C:\TBC\Sem2\POP_Project\Weekly Assignments\Practical4> 

```

4. Distinct Number

```
#include <stdio.h>
#include <stdbool.h>

int main()
{
    int size=10, input_num[size], num_count=0,
    d_num[num_count] ;
    bool dist;

    //taking input
    printf("Enter %d numbers: ",size);
    for(int i=0; i<size; i++)
    {
        scanf("%d",&input_num[i]);
    }

    for(int i=0; i<size; i++)
    {
        dist = true;
        //2nd loop to check if number already presents
        //looping until i's size
        //(looping with size, it'll always return false cause it'll be
        comparing all with itself)
        for(int j=0;j<i;j++)
        {
            if(input_num[i]==input_num[j])
            {
                //means there's a duplicate value
                dist=false;
            }
        }
    }
}
```

```

        //if one duplicate value found breaks
        break;
    }
}

if (dist==true)
{
    d_num[num_count]=input_num[i];
    num_count++;
}

}

printf("The number of distinct number: %d\n",num_count);
printf("The distinct numbers are: ");
for(int i=0; i<num_count;i++)
{
    printf("%d ",d_num[i]);
}

return 0;
}

```

Output:

```

m2\POP_Project\Weekly Assignments\Practical4\
tinctNumbers.c -o printDistinctNumbers } ; it
bers }
Enter 10 numbers: 1 2 3 2 1 6 3 4 5 2
The number of distinct number: 6
The distinct numbers are: 1 2 3 6 4 5
PS C:\TBC\Sem2\POP_Project\Weekly Assignments

```

5. Minimum Number:

```
#include <stdio.h>

int main()
{
    int size =10;
    float input_nums[size], mini_num;
    printf("Enter %d numbers: ",size);
    for(int i=0; i<size; i++)
    {
        scanf("%f",&input_nums[i]);
        //supposing mininum number is in index 0
        mini_num=input_nums[0];
    }

    for(int i=0; i<size;i++)
    {
        if(mini_num>input_nums[i])
        {
            mini_num = input_nums[i];
        }
    }

    printf("The minimum number is: %.1f",mini_num);

    return 0;
}
```

Output:

```
m2\POP_Project\Weekly Assignments\Practical4\" ; if ($?) { gcc
_num.c -o smallest_num } ; if ($?) { .\smallest_num }
Enter 10 numbers: 1.9 2.5 3.7 2 1.5 6 3 4 5 2
The minimum number is: 1.5
PS C:\TBC\Sem2\POP Project\Weekly Assignments\Practical4>
```

6. Column Sum:

```
#include <stdio.h>

int main()
{
    int row=2, col=3;
    float matrix[row][col],sum[col];

    printf("Enter a 3-by-4 matrix row by row: \n");
    for(int i=0;i<row;i++)
    {
        for(int j=0; j<col;j++)
        {
            scanf("%f",&matrix[i][j]);
        }
    }

    //looping through columns of rows
    for(int i=0; i<col;i++)
    {
        for(int j=0; j<row;j++)
        {
            //adding the col of the rows
            sum[i]+= matrix[j][i];
        }
    }

    //printing the sum of each column
    for(int i=0;i<col; i++)
    {
```

```
        printf("Sum of the elements at column %d is %.1f\n",i,sum[i]);
    }

    return 0;
}
```

Output:

```
● Enter a 3-by-4 matrix row by row:
1.5 2 3 4
5.5 6 7 8
9.5 1 3 1
Sum of the elements at column 0 is 16.5
Sum of the elements at column 1 is 9.0
Sum of the elements at column 2 is 13.0
Sum of the elements at column 3 is 13.0
○ PS C:\TBC\Sem2\POP_Project\Weekly Assignments\Practical4> █
```


7. Employee Working hour:

```
#include <stdio.h>

int main()
{
    int total_hours[8]={0},total_emp[8],temp_hours,temp_emp;
    //8 employees working hours (7days)
    //when values are given, it should be done during initialization
    int w_hours[8][7]=
    {
        {2,4,3,4,5,8,8},
        {7,3,4,3,3,4,4},
        {3,3,4,3,3,2,2},
        {9,3,4,7,3,4,1},
        {3,5,4,3,6,3,8},
        {3,4,4,6,3,4,4},
        {3,7,4,8,3,8,4},
        {6,3,5,9,2,7,9}
    };

    for (int i=0; i<8;i++)
    {
        for(int j=0; j<7;j++)
        {
            total_hours[i]+=w_hours[i][j];
        }
        //printing emp with working hours
        total_emp[i]=i;
        //printf("Employee %d total working hour:
        %d\n",i,total_hours[i]);
    }
}
```

```

}

printf("\n");

//Ordering empp with their working hours in descending order
for(int i=0;i<8;i++)
{
    for(int j =0; j<8;j++)
    {
        if(total_hours[i]>total_hours[j])
        {
            temp_hours = total_hours[j];
            total_hours[j] = total_hours[i];
            total_hours[i]= temp_hours;

            temp_emp=total_emp[j];
            total_emp[j]= total_emp[i];
            total_emp[i]=temp_emp;
        }
    }
}

//printing in descending order
for(int i=0;i<8;i++)
{
    printf("Employee %d total working hour:
%d\n",total_emp[i],total_hours[i]);
}

return 0;

```

```
}
```

Output:

```
m2\POP_Project\Weekly Assignments\Practical4\" ; if ($?) {  
ly_hours.c -o emp_weekly_hours } ; if ($?) { .\emp_weekly_  
  
Employee 7 total working hour: 41  
Employee 6 total working hour: 37  
Employee 0 total working hour: 34  
Employee 4 total working hour: 32  
Employee 3 total working hour: 31  
Employee 1 total working hour: 28  
Employee 5 total working hour: 28  
Employee 2 total working hour: 20  
PS C:\TBC\Sem2\POP_Project\Weekly Assignments\Practical4>
```