

***Use the < Operator to Redirect Standard Input***

File redirection is generally known as I/O redirection on UNIX based systems, which allows the user to redefine where standard input comes from, or standard output goes. < operator is used to change where the standard input comes from. This method can be useful for taking user input from the file contents using scanf function directly. In the below example the file Redirect.c will take the roll no and names of ten students from students.txt and will store in array of structure s1. It will then print the output on console.

File Redirect.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Student {
    int roll_no;
    char name[20];
};
int main(int argc, char *argv[])
{
    int i;
    struct Student s1[20];

    for(i=0;i<10;i++)
    {
        scanf("%d",&s1[i].roll_no);
        scanf("%s",s1[i].name);
    }
    for(i=0;i<10;i++)
    {
        printf("%d\t",s1[i].roll_no);
        printf("%s\n",s1[i].name);
    }
    exit(EXIT_SUCCESS);
}
```

File students.txt

```
1 Rohit
2 Mohit
3 Hardik
4 Shashank
5 Amit
6 Revant
7 Vishal
8 Dilip
9 Tarun
10 Ashwin
```

Sample Command:

```
./a.out < students.txt
```

Output:

```
1 Rohit
2 Mohit
3 Hardik
4 Shashank
5 Amit
6 Revant
7 Vishal
8 Dilip
9 Tarun
10 Ashwin
```

- 1 "Do Indian parents prefer giving shorter names over longer ones to their children?" You will write a *well-commented* and modular C program to help get a possible answer to this question, using data from a given sample population. You will use the following structure definition in your program:

```
typedef struct
{
    char fname[70]; /* first name (a single word) */
    int freq; /* how many people in the sample population have this
    name */
    int len; /* stores the length of the name */
} NAME;
```

Given a sample size of **N** individuals, and a positive integer **threshold** denoting a length, your program should do these tasks:

- (a) Obtain data for **N** individuals and store it appropriately using the function:

```
void getData(NAME arr[]);
```

- (b) Calculate and print the percentage of people in the sample population whose first names are shorter than or equal to the threshold length, using another function:

```
float findPercent(NAME arr[], int threshold);
```

The file `names.txt` has sample data – each line (except the first one) containing a unique first name and the frequency of its occurrence in the sample population. For example, in the sample data shown below, there are six people with the name Abhinav, four with Abhimanyu, two with Abhay and one each of Abhas, Abhay, Abhijeet and Abhijit:

ABHAS	1
ABHAY	2
ABHIJEET	1
ABHIJIT	1
ABHIMANYU	4
ABHINAV	6

The first line of the file contains two numbers – the number of unique names (stored one per row) and the value for threshold.

2. Write a C program that computes the transpose of a 2D array using pointers and dynamic memory allocation.

3. This program involves taking input for **m** rows of a matrix, but each row having a varying number of integer elements. Take a look at 12.6.txt to get an idea of this matrix with varying number of columns for each row. The first row contains the number of rows. The first element of each row contains the number of columns of that particular row, followed by the actual elements themselves.

Now declare a suitable array of pointers, dynamically allocate memory for each row, read the elements (including the number of elements as the first entry of each row), and then display the entire matrix on the screen. Verify to make sure that your output is the same as the matrix entries given in the data file 12.6.txt