You are given an n x n 2D matrix representing an image. Rotate the image by 90 degrees (clockwise).

Input

 [   [1,2,3],

     [4,5,6],

     [7,8,9]

     ]

#include<stdio.h>

void rotate(int arr[][3],int n)

{

for(int i=0;i<n;i++)

{

for(int j=i;j<n;j++)

{

int temp=arr[i][j];

arr[i][j]=arr[j][i];

arr[j][i]=temp;

}

}

for(int i=0;i<n;i++)

{

for(int j=0;j<n/2;j++)

{

int temp=arr[i][j];

arr[i][j]=arr[i][n-j-1];

arr[i][n-j-1]=temp;

}

}

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

printf("%d ",arr[i][j]);

}

printf("\n");

}

}

int main()

{

int matrix[3][3]={

{1,2,3},

{4,5,6},

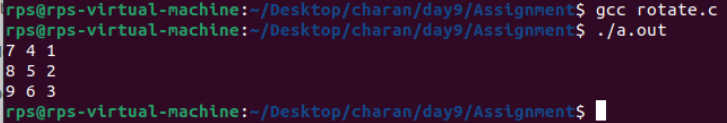
{7,8,9}

};

int size=3;

rotate(matrix,size);

}



2)You are assigned the task of developing a C program that enables users to input a question and specify the filename where the question will be saved. However, there's a constraint that one file cannot have more than 10 questions.

Program Requirements:

The program should start by prompting the user to input a que

After the question is entered, the program should prompt the user to specify the filstion.ename where the question will be saved.

Both the question and the filename should be inputted using the command line.

The program should open the specified file in append mode. If the file doesn't exist, it should create it.

Each question should be stored in the file on a new line.

The program should enforce the constraint that one file cannot have more than 10 questions. If this limit is reached, the program should inform the user and exit.

After successfully saving the question, the program should display a confirmation message.

Constraints:

The maximum length of a question is 100 characters.

The maximum length of a filename is 50 characters.

Each file can have a maximum of 10 questions

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define MAX\_QUES\_LENGTH 100

#define MAX\_FILENAME\_LENGTH 50

#define MAX\_QUESTIONS 10

int countLines(FILE \*file)

{

int count =0;

char ch;

while((ch = fgetc(file)) != EOF)

{

if(ch == '\n')

{

count++;

}

}

rewind(file);

return count;

}

int main(int argc, char \*argv[]){

if(argc != 3)

{

fprintf(stderr,"Usage: %s <question> <filename>\n",argv[0]);

return 1;

}

char \*question = argv[1];

char \*filename = argv[2];

if(strlen(filename) > MAX\_QUES\_LENGTH){

fprintf(stderr,"Error:Question length exceeds %d characters.\n",MAX\_QUES\_LENGTH);

return 1;

}

if(strlen(filename) > MAX\_FILENAME\_LENGTH)

{

fprintf(stderr,"Error:Filename length exceeds %d characters.\n",MAX\_FILENAME\_LENGTH);

return 1;

}

FILE \*file =fopen(filename,"a+");

if(file == NULL){

perror("Error opening file");

return 1;

}

int quesCount = countLines(file);

if(quesCount >= MAX\_QUESTIONS)

{

fprintf(stderr,"Error: File \' %s\' already contains %d questions.Cannot add more. \n",filename,MAX\_QUESTIONS);

fclose(file);

return 1;

}

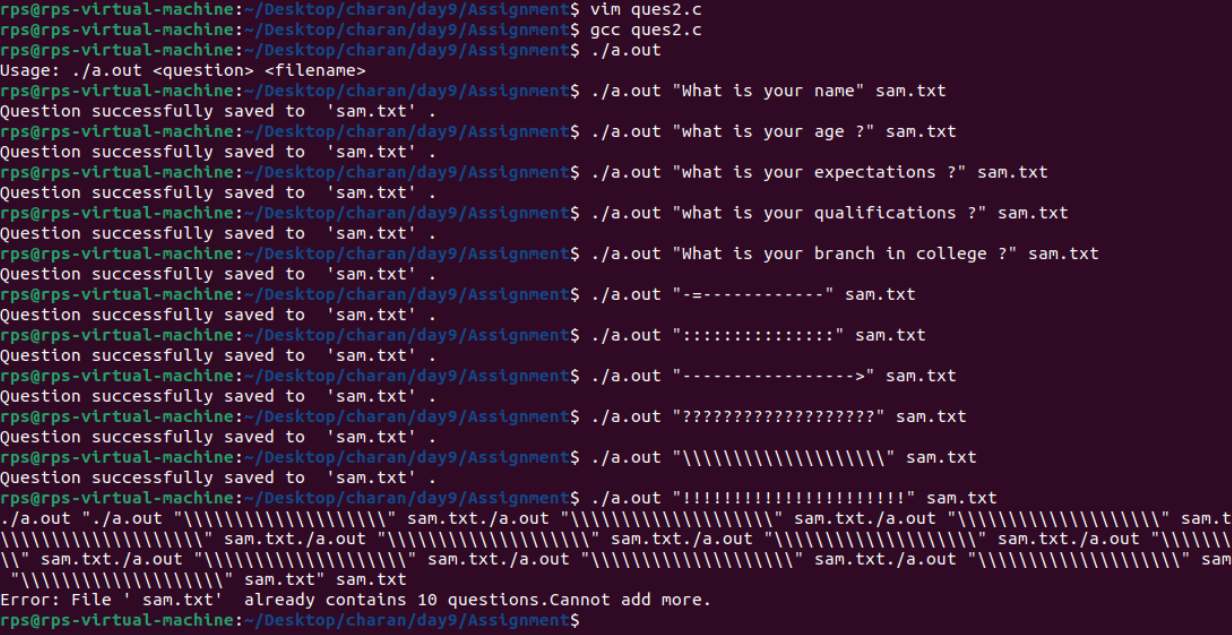
fprintf(file,"%s\n",question);

fclose(file);

printf("Question successfully saved to \'%s\' .\n",filename);

return 0;

}

3.1)check if number is perfect number or not ?

3.2) Given an array of integers nums and an integer target, return the indices of the two numbers such that they add up to target. You may assume that each input would have exactly one solution, and you may not use the same element twice

#include<stdio.h>

#include<stdbool.h>

int main()

{

int num;bool temp;

printf("Enter number : ");

scanf("%d",&num);

for(int i=2;i<=num/2;i++)

{

if((num/i)==i)

{

temp=true;

}

}

if(temp)

{

printf("Perfect Number\n");

return 0;

}

else

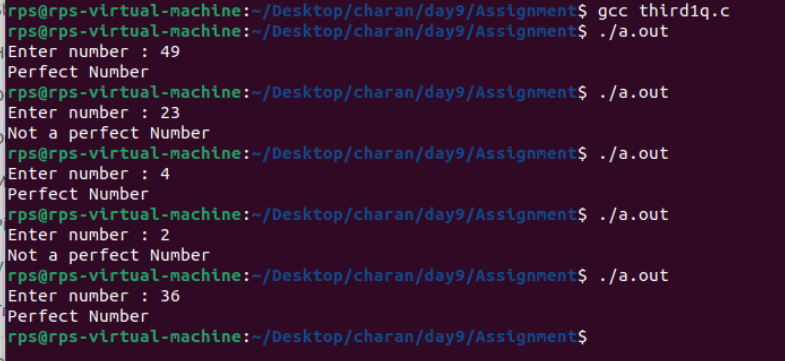
{

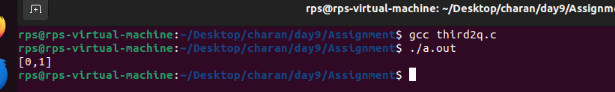
printf("Not a perfect Number\n");

return 0;

}

}





3.2)#include<stdio.h>

void targetSum(int\* arr,int target){

int length=sizeof(arr)/sizeof(arr[0]);

for(int i=0;i<length;i++)

{

for(int j=i+1;j<length;j++)

{

if((arr[i]+arr[j])==target)

{

printf("[%d,%d]\n",i,j);

return;

}

}

}

}

int main()

{

int array[]={2,7,11,15};

int target =9;

targetSum(array,target);

}

Given a non-empty array of integers nums, where every element appears twice except for one, find that single one.

input : nums = [4, 1, 2, 1, 2]

output: 4

n the array [4, 1, 2, 1, 2], every element appears twice except 4.

To find the single number that appears only once, we can use the XOR bitwise operator due to its properties:

* a ^ a = 0: XORing a number with itself results in 0.
* a ^ 0 = a: XORing a number with 0 results in the number itself.

By XORing all elements in the array:4

(4 ^ 4) ^ (1 ^ 1) ^ (2 ^ 2) = 0 ^ 0 ^ 0 ^ 0 ^ 4 = 4

(Tip :Revise Exor truth table)

#include<stdio.h>

int Findnum(int \*nums,int length)

{

int temp=0;

for(int i=0;i<length;i++)

{

temp ^= nums[i];

}

return temp;

}

int main()

{

int nums[]={4,1,2,1,2};

int n=sizeof(nums)/sizeof(nums[0]);

printf("The value is %d \n",Findnum(nums,n));

}

