{
}

Introduction to Programming;
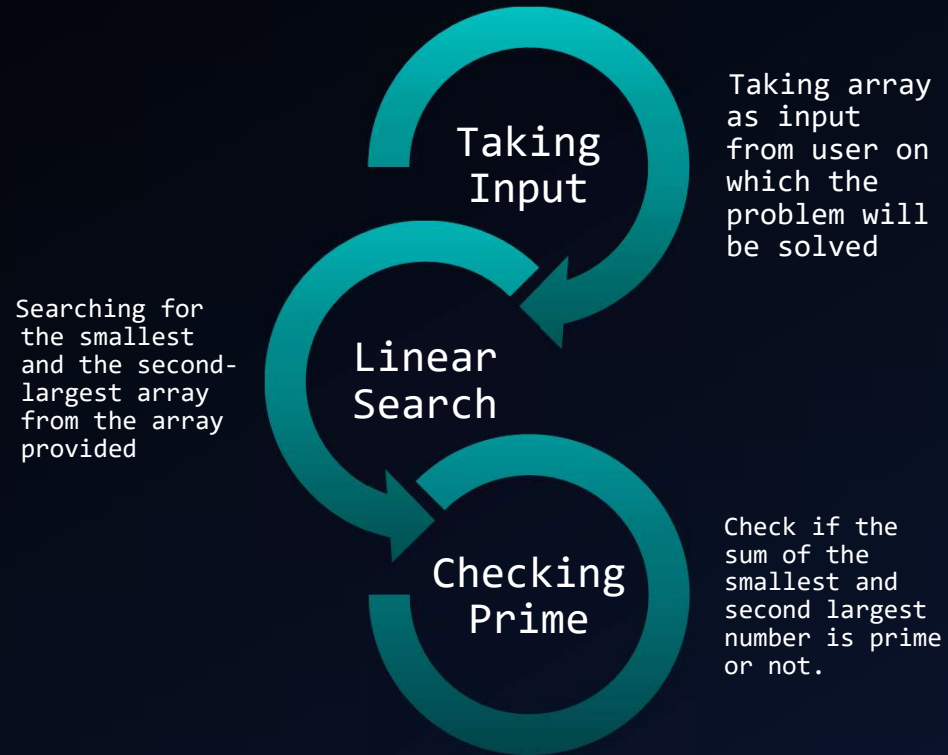
ASSIGNMENT-1

BY GROUP-1

The Problem ...

Given an unsorted array of n numbers, find the sum of the second largest and smallest number and then check whether it is prime or not.

The Solution

WE PROPOSE

Our Approach

Taking
Input

Taking array
as input
from user on
which the
problem will
be solved

Searching for
the smallest
and the second-
largest array
from the array
provided

Linear
Search

Checking
Prime

Check if the
sum of the
smallest and
second largest
number is prime
or not.

TAKING INPUT

1. First size of the array is asked.

2. Then the individual element is asked, index-wise.

1
```c
int n;
printf("Enter the size of
   the array : ");
scanf("%d",&n);
```

2
```c
int a[n];
for(int i=0;i<n;i++){
printf("Enter the element at
   the index %d of the array :
   ", i);
scanf("%d",&a[i]);
}
```

```
int min, max, max2;
max2 = max = -2147483647;
min = 2147483647;

for(int i=0; i<(n); i++){
    if(min>a[i]){
        min = a[i];
    }if(max<a[i]){
        max2 = max;
        max = a[i];
    } else if(max2<a[i]){
        max2 = a[i];
    }
}
```

LINEAR SEARCH

To find the smallest and the second largest number from the array.

The Linear Search

The program starts with three variables, one for minimum, one for maximum, and one for second maximum. The maximum and second maximum are initialized to the smallest integer possible and the minimum is initialized to the largest integer possible in the C programming language. Then they are compared to each element of the array and checked if there is any element less than the number stored in minimum and any element bigger than the number stored in the maximum variable. Every time the value of maximum is changed, its previous value is assigned to the second-maximum variable. Also, it checks if there is any number smaller than the maximum but bigger than the second-maximum and then changes its value accordingly.
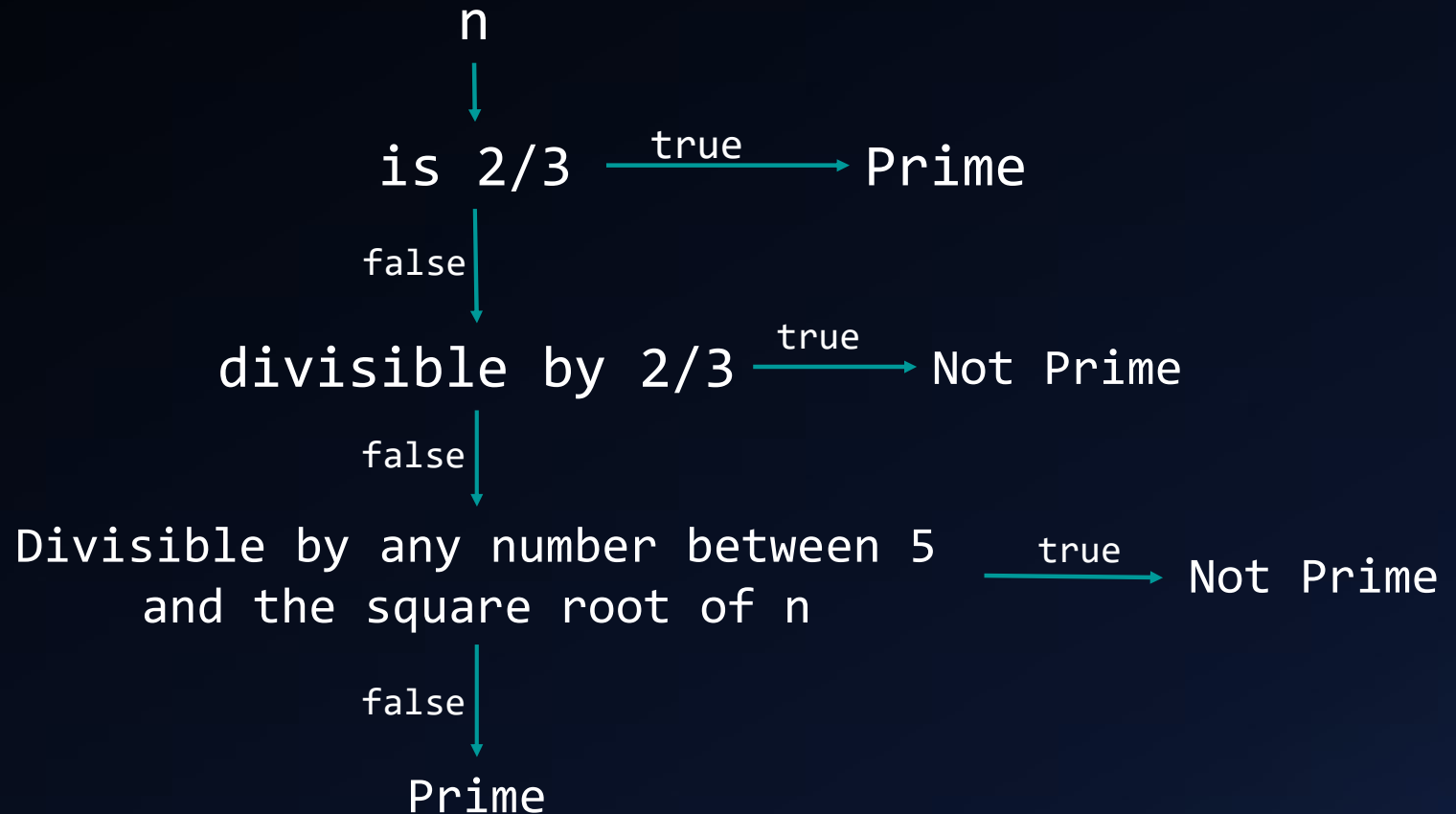
CHECKING PRIME

To check whether the sum of the smallest and the second largest number is a prime number or not.

```cpp
bool checkPrime(int n)
{
    if ((n == 2)||( n == 3)){
        return true;
    }
    else if (((n % 2) == 0)||((n % 3) == 0)||((n<=0))){
        return false;
    }
    else {
        for(int i = 5; i <= sqrt(n); i++){
            if (n % i == 0){
                return false;
            }
        }
    }
    return true;
}
```

Checking Prime

for an integer 'n'

n

is 2/3 ——true——> Prime

false

divisible by 2/3 ——true——> Not Prime

false

Divisible by any number between 5 and the square root of n ——true——> Not Prime

false

Prime

Test Cases

SN	Array Used	Output (Sum)	Runtime (without Input)
1.	85, 45, -5, 12, -7, 155, 69, 115, 58, 59	Not Prime(108)	0.898
2.	12, 6, 45, 13, 9, 8	Prime(19)	0.247
3.	-69, 70, 169, 12, 13	Neither(1)	0.254

Most Important Code:

```c
int greet(){
    printf("Thank You!"); //we need marks
    return 0;
}
```

Output:

```
PS E:\ITP Assignment> ./greet.exe
Thank You!
```

From Group-1

Mohd. Wael Abrar
(IEC2021001)

Anurag Jain
(IEC2021002)

Abhishek Rana
(IEC2021003)

Ayush Kumar Gupta
(IEC2021004)