LAPORAN PRAKTIKUM

Algoritma Pemrograman

MODUL 10

If-else

Disusun oleh:

Cofa Xavier Marvel

109082500001

S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

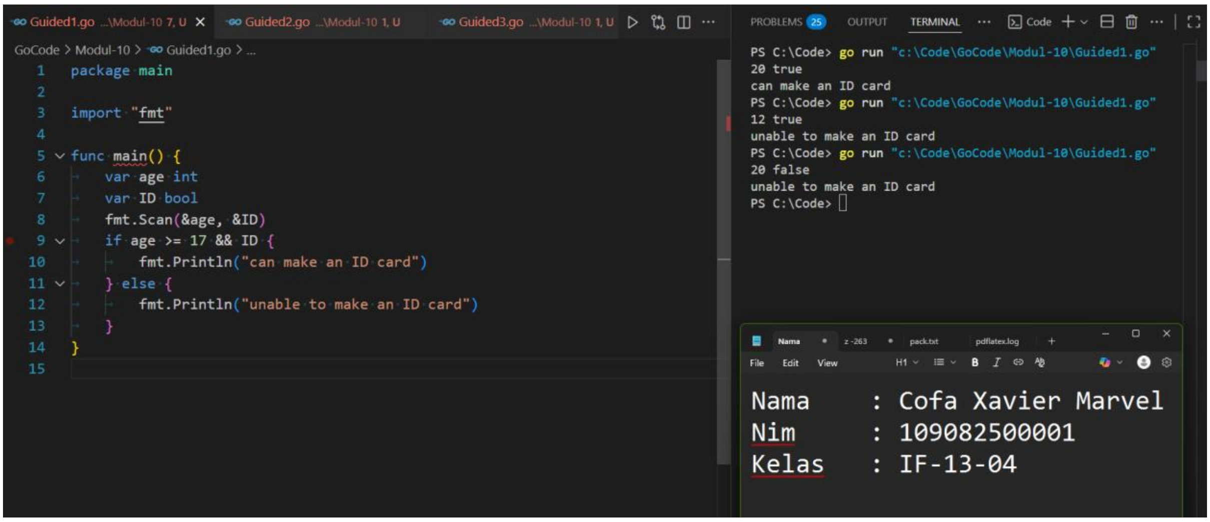
L

1. **Guided 1**

   Source Code

```go
package main

import "fmt"

func main() {
    var age int
    var famcard bool
    fmt.Scan(&age, &famcard)
    if age >= 17 && famcard {
        fmt.Println("can make an ID card")
    } else {
        fmt.Println("unable to make an ID card")
    }
}
```

   Screenshoot program



Guided1.go ...\Modul-10 7, U
Guided2.go ...\Modul-10 1, U
Guided3.go ...\Modul-10 1, U
PROBLEMS 25
OUTPUT
TERMINAL
Code
GoCode > Modul-10 > Guided1.go > ...
Nama
z -263
pack.txt
pdflatex.log
File
Edit
View

```go
package main

import "fmt"

func main() {
    var age int
    var ID bool
    fmt.Scan(&age, &ID)
    if age >= 17 && ID {
        fmt.Println("can make an ID card")
    } else {
        fmt.Println("unable to make an ID card")
    }
}
```

```
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided1.go"
20 true
can make an ID card
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided1.go"
12 true
unable to make an ID card
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided1.go"
20 false
unable to make an ID card
PS C:\Code>
```

```
Nama    : Cofa Xavier Marvel
Nim     : 109082500001
Kelas   : IF-13-04
```

   Deskripsi program

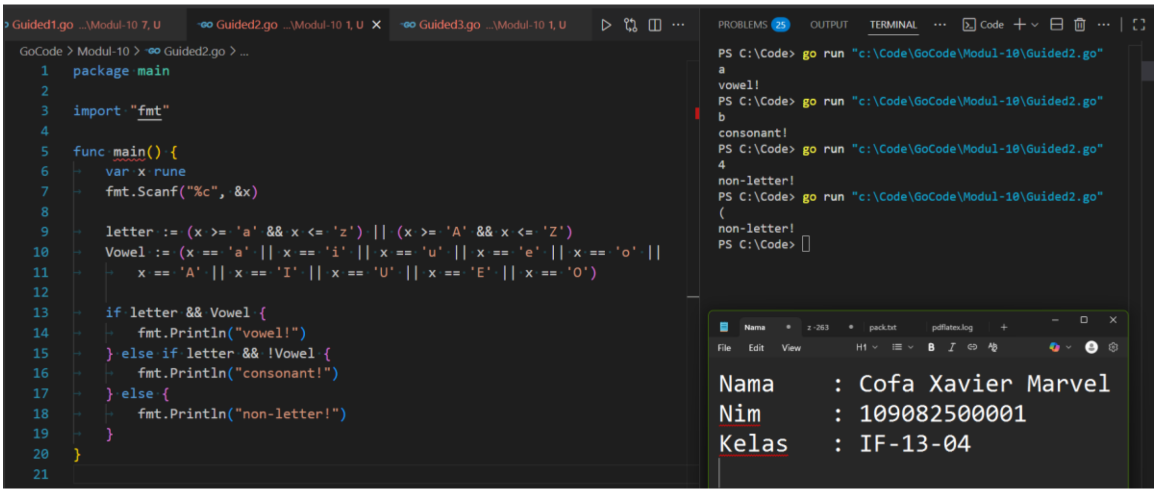
   This program uses an if else statement where if age is greater or equal to 17 and famcard is true then print “can make an ID Card” else print “unable to make an ID card”

2. **Guided 2**

Source Code

```go
ppackage main

import "fmt"

func main() {
    var x rune
    fmt.Scanf("%c", &x)

    letter := (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
    Vowel := (x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o' ||
        x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O')

    if letter && Vowel {
        fmt.Println("vowel!")
    } else if letter && !Vowel {
        fmt.Println("consonant!")
    } else {
        fmt.Println("non-letter!")
    }
}
```

Screenshoot program


Guided1.go ...\Modul-10 7, U
Guided2.go ...\Modul-10 1, U
Guided3.go ...\Modul-10 1, U
GoCode > Modul-10 > Guided2.go > ...
PROBLEMS 25
OUTPUT
TERMINAL
Code
Nama
z -263
pack.txt
pdflatex.log
File
Edit
View
H1

```go
package main

import "fmt"

func main() {
    var x rune
    fmt.Scanf("%c", &x)

    letter := (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
    Vowel := (x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o' ||
        x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O')

    if letter && Vowel {
        fmt.Println("vowel!")
    } else if letter && !Vowel {
        fmt.Println("consonant!")
    } else {
        fmt.Println("non-letter!")
    }
}
```

```
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided2.go"
a
vowel!
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided2.go"
b
consonant!
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided2.go"
4
non-letter!
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided2.go"
(
non-letter!
PS C:\Code>
```

Nama   : Cofa Xavier Marvel
Nim   : 109082500001
Kelas   : IF-13-04

Deskripsi program

This program finds whether the rune given is either a vowel, consonant or non-letter.

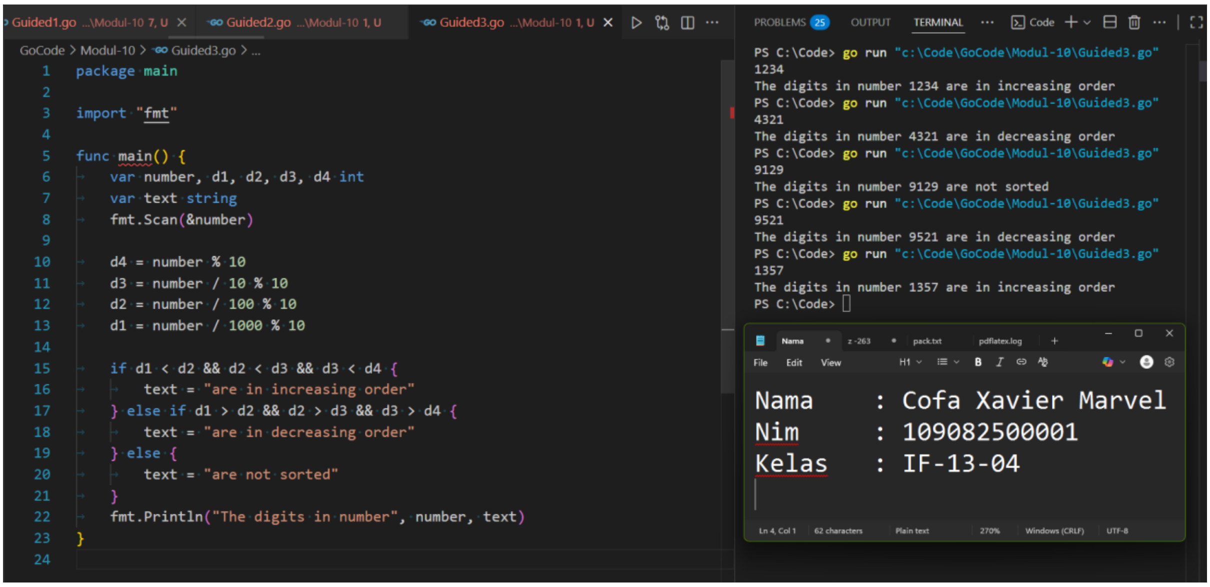
It does this using an if-else statement if is letter and is vowel then it is a vowel else if is letter and not vowel then it is consonant else non-letter

3. Guided 3

Source Code

```go
package main

import "fmt"

func main() {
    var number, d1, d2, d3, d4 int
    var text string
    fmt.Scan(&number)

    d4 = number % 10
    d3 = number / 10 % 10
    d2 = number / 100 % 10
    d1 = number / 1000 % 10

    if d1 < d2 && d2 < d3 && d3 < d4 {
        text = "are in increasing order"
    } else if d1 > d2 && d2 > d3 && d3 > d4 {
        text = "are in decreasing order"
    } else {
        text = "are not sorted"
    }
    fmt.Println("The digits in number", number, text)
}
```

Screenshoot program



```go
package main

import "fmt"

func main() {
	var number, d1, d2, d3, d4 int
	var text string
	fmt.Scan(&number)

	d4 = number % 10
	d3 = number / 10 % 10
	d2 = number / 100 % 10
	d1 = number / 1000 % 10

	if d1 < d2 && d2 < d3 && d3 < d4 {
		text = "are in increasing order"
	} else if d1 > d2 && d2 > d3 && d3 > d4 {
		text = "are in decreasing order"
	} else {
		text = "are not sorted"
	}
	fmt.Println("The digits in number", number, text)
}
```

```
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided3.go"
1234
The digits in number 1234 are in increasing order
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided3.go"
4321
The digits in number 4321 are in decreasing order
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided3.go"
9129
The digits in number 9129 are not sorted
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided3.go"
9521
The digits in number 9521 are in decreasing order
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Guided3.go"
1357
The digits in number 1357 are in increasing order
PS C:\Code>
```

```
Nama     : Cofa Xavier Marvel
Nim      : 109082500001
Kelas    : IF-13-04
```

Deskripsi program

This program finds if the 4-digit number is ordered descending, ascending or not ordered. Using an if-else statement if digit1 is smaller than digit2 and digit 2 is smaller than digit3 and digit3 is smaller than digit4 then it is ascending order the inverse is descending order if both statements are false then not ordered

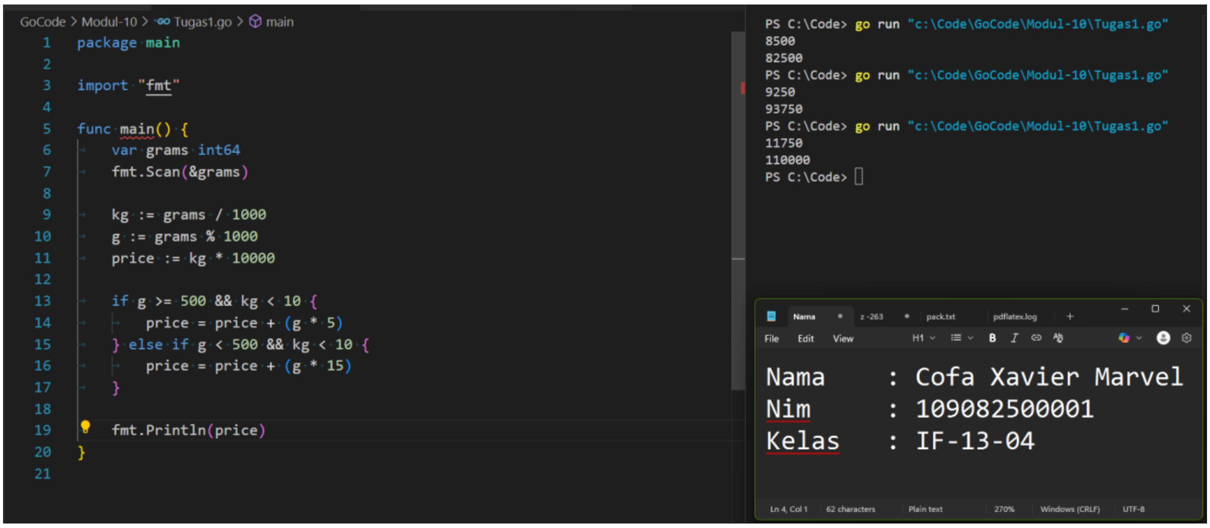
TUGAS

Tugas 1

Source code

```go
package main

import "fmt"

func main() {
    var grams int64
    fmt.Scan(&grams)

    kg := grams / 1000
    g := grams % 1000
    price := kg * 10000

    if g >= 500 && kg < 10 {
        price = price + (g * 5)
    } else if g < 500 && kg < 10 {
        price = price + (g * 15)
    }

    fmt.Println(price)
}
```

Screenshoot program



GoCode > Modul-10 > Tugas1.go > main

```go
package main

import "fmt"

func main() {
	var grams int64
	fmt.Scan(&grams)

	kg := grams / 1000
	g := grams % 1000
	price := kg * 10000

	if g >= 500 && kg < 10 {
		price = price + (g * 5)
	} else if g < 500 && kg < 10 {
		price = price + (g * 15)
	}

	fmt.Println(price)
}
```

```
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Tugas1.go"
8500
82500
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Tugas1.go"
9250
93750
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Tugas1.go"
11750
110000
PS C:\Code> 
```

```
Nama     : Cofa Xavier Marvel
Nim      : 109082500001
Kelas    : IF-13-04
```

Deskripsi program

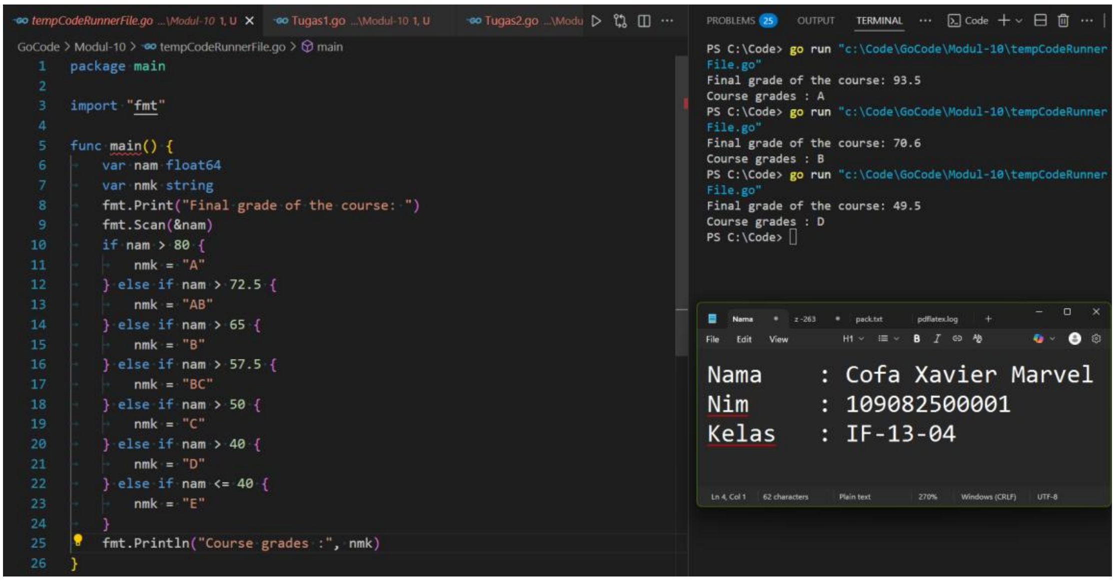
This program calculate the shipping fee with the stipulation that shipping service fee is Rp. 10,000 per kg. If the remaining weight is not less than 500 grams, then the additional shipping fee is only Rp. 5 per gram. But if it is less than 500 grams, then an additional fee will be charged of IDR 15 per gram. The remaining weight (which is less than 1kg) is free of charge if the total weight is more than 10kg

Tugas 2

Source code

```go
package main

import "fmt"

func main() {
    var nam float64
    var nmk string
    fmt.Print("Final grade of the course: ")
    fmt.Scan(&nam)
    if nam > 80 {
        nmk = "A"
    } else if nam > 72.5 {
        nmk = "AB"
    } else if nam > 65 {
        nmk = "B"
    } else if nam > 57.5 {
        nmk = "BC"
    } else if nam > 50 {
        nmk = "C"
    } else if nam > 40 {
        nmk = "D"
    } else if nam <= 40 {
        nmk = "E"
    }
    fmt.Println("Course grades :", nmk)
}
```

Screenshoot program



tempCodeRunnerFile.go ...\Modul-10 1, U
Tugas1.go ...\Modul-10 1, U
Tugas2.go ...\Modu
GoCode > Modul-10 > tempCodeRunnerFile.go > main
PROBLEMS 25
OUTPUT
TERMINAL
Code

```
package main

import "fmt"

func main() {
    var nam float64
    var nmk string
    fmt.Print("Final grade of the course: ")
    fmt.Scan(&nam)
    if nam > 80 {
        nmk = "A"
    } else if nam > 72.5 {
        nmk = "AB"
    } else if nam > 65 {
        nmk = "B"
    } else if nam > 57.5 {
        nmk = "BC"
    } else if nam > 50 {
        nmk = "C"
    } else if nam > 40 {
        nmk = "D"
    } else if nam <= 40 {
        nmk = "E"
    }
    fmt.Println("Course grades :", nmk)
}
```

```
PS C:\Code> go run "c:\Code\GoCode\Modul-10\tempCodeRunnerFile.go"
Final grade of the course: 93.5
Course grades : A
PS C:\Code> go run "c:\Code\GoCode\Modul-10\tempCodeRunnerFile.go"
Final grade of the course: 70.6
Course grades : B
PS C:\Code> go run "c:\Code\GoCode\Modul-10\tempCodeRunnerFile.go"
Final grade of the course: 49.5
Course grades : D
PS C:\Code>
```

```
Nama   : Cofa Xavier Marvel
Nim    : 109082500001
Kelas  : IF-13-04
```

Deskripsi program

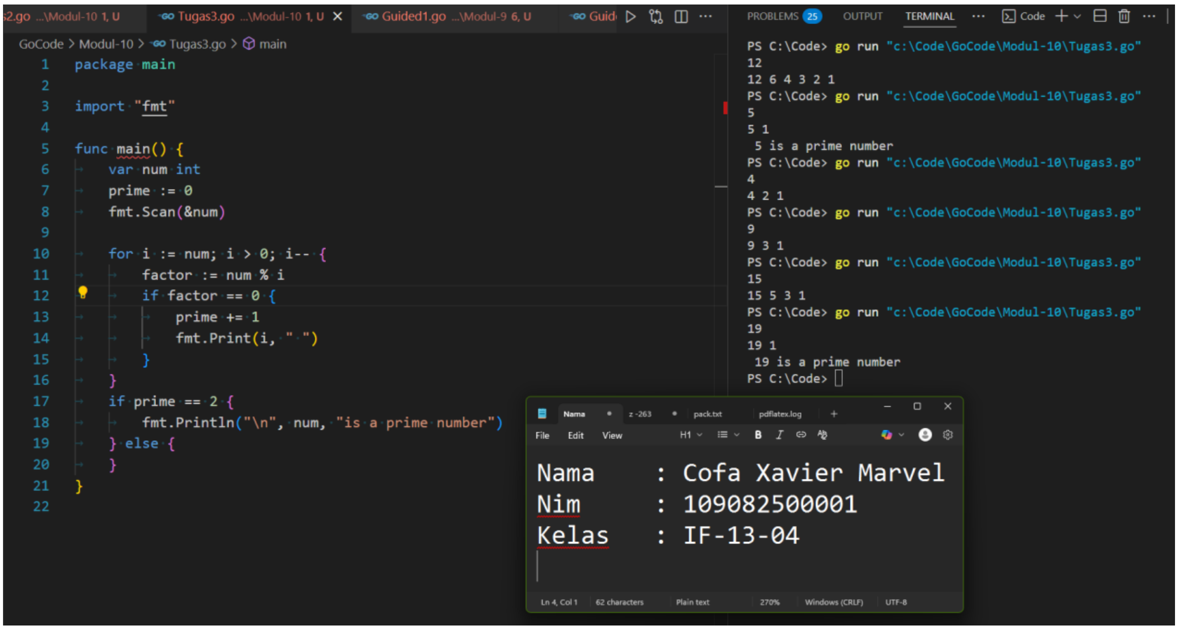
The program accepts as input a real number that represents NAM. The program finds NMK and displays it.

Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var num int
    prime := 0
    fmt.Scan(&num)

    for i := num; i > 0; i-- {
        factor := num % i
        if factor == 0 {
            prime += 1
            fmt.Print(i, " ")
        }
    }
    if prime == 2 {
        fmt.Println("\n", num, "is a prime number")
    } else {
        fmt.Println("\n", num, "is not a prime number")
    }
}
```

Screenshoot program



```
package main

import "fmt"

func main() {
    var num int
    prime := 0
    fmt.Scan(&num)

    for i := num; i > 0; i-- {
        factor := num % i
        if factor == 0 {
            prime += 1
            fmt.Print(i, " ")
        }
    }
    if prime == 2 {
        fmt.Println("\n", num, "is a prime number")
    } else {
    }
}
```

```
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Tugas3.go"
12
12 6 4 3 2 1
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Tugas3.go"
5
5 1
 5 is a prime number
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Tugas3.go"
4
4 2 1
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Tugas3.go"
9
9 3 1
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Tugas3.go"
15
15 5 3 1
PS C:\Code> go run "c:\Code\GoCode\Modul-10\Tugas3.go"
19
19 1
 19 is a prime number
PS C:\Code>
```

```
Nama    : Cofa Xavier Marvel
Nim     : 109082500001
Kelas   : IF-13-04
```

Tugas3.go ...\Modul-10 1, U
Guided1.go ...\Modul-9 6, U
GoCode > Modul-10 > Tugas3.go > main
PROBLEMS 25
OUTPUT
TERMINAL
Code
File
Edit
View

Deskripsi program

The program is given an integer num and num > 1. The program finds and display all factors of the number Then, the program determines whether b is a prime number.