COMPUTER SCIENCE ASSIGNMENT 2

Name: Ruhul Amin

Branch: Electronics and Communication Engineering

Semester: 1st Semester Roll No.: B25EC014

1. What is C language? Who developed C language?

Answer:

C is general-purpose, high-level programming language that also provides low-level features, making it very powerful for systems programming.

C was developed in the early 1970s by Dennis Ritchie at Bell Labs (AT&T, USA).

2. What do you mean by low level, middle level and high level language? Give example for each.

Answer:

Low-level language: Closer to machine code, hard for humans, but very fast and hardware-specific. Example: Assembly language.

Middle-level language: Bridges gap between low and high level. Can do system-level tasks (like hardware access) and application-level tasks. Example: C.

High-level language: Easy for humans to read/write, abstracted from hardware, portable across systems. Example: Python, Java, etc

3. Short note on structure oriented, object oriented and nonstructure oriented programming language?

Answer:

Non-structured language: They have no clear program structure, uses jumps (like goto), harder to manage for large programs. For example: BASIC, Assembly, etc.

Structure-oriented language: Programs are divided into functions/modules, uses control structures (loops, if-else), easier to read and maintain. For example: C, Pascal, etc.

Object-oriented language: Based on objects (data + functions together), supports concepts like classes, inheritance'[

4. What do you mean by compiler, assembler, and interpreter? Answer:

Compiler: Translates the whole program (from high level to machine code) at once. Fast execution, but errors are shown after full compilation.

Assembler: Converts assembly language into machine code.

Interpreter: Translate and executes program line by line. Slower, but errors are shown immediately.

5. What is the only function all C programs must contain? What punctuation is used to signal the beginning and end of code blocks? What punctuation ends most lines of C code?

Answer:C

main() is the function that all c programs must contain. {} is used to begin and end any code block. Most statements ends with ':' in c code.

6. What is data type? What is the size (no of bytes) of int, float and char data type?

Answer:

A data type is the type of data a variable can store. int and float has a size of 4 bytes while char has size of 1 byte.

7. Is C programming language, structure oriented or procedure oriented language?

Answer:

C is a procedure-oriented programming language.

8. Difference between a variable and a constant in C and how do we define them, show with example?

Answer:

A variable refers to the pointer that points at a certain area in the memory whereas a constant is a data that has been stored in that area of the memory. We first have to define the variable and only after that we assign a constant value to it. Like the example written as follows:

```
int number;  //defining variable
number = 3;  //defining the constant
```

9. What do you mean by declaration and initialization?

Answer:

The process of telling the compiler about a variable's name and type, whereas, the process assigning the initial value to the variable is called initialization.

10. Can we run a program without main function? What is library function?

Answer:

No we can't run a program without main() function. A library function refers to the functions predefined function provided by the c libraries to perform common tasks.

11. What do you mean by "stdio.h"? Why do we write " return 0" in main function?

Answer:

stdio.h is a library function of input/output functions like printf()
and scanf().

12. What is compile time error and run time error?

Answer:

Compile-time errors are the errors detected by the compiler while translating code while run-time error are the errors caused while running the program.

13. What are logical errors and how does it differ from syntax errors?

Answer:

if the program runs and gives an output but the output wasn't the intended output, it is called a logical error. A syntax error refers to error that are caused when the code is written in an invalid manner or the rules of c were broken while writing the code.

14. What do you understand by identifiers and keywords? Answer:

An identifier is a name given to a variable, functions, arrays, etc, by the programmer. A keyword refers to the 32 predefined in c with a special meaning.

15. What are arithmetic operators? Modulus operator (%) can be used in which data type?

Answer:

Arithmetic operators like +, -, %, /, *, etc are used to perform basic mathematical operations in C. Modulus operator can only be used with int data type

16. Arithmetic and logical expressions are evaluated from which direction of a code line?

Answer:

Arithmetic and logical expressions are evaluated from left to right direction of the code if the precedence is the same, otherwise the operator with higher precedence is given the first preference and so on.

17. What are logical operators, conditional operators and bitwise operators?

Answer:

Logical operators are used to handle true or false conditions.

Conditional operators are used to compare two data, for example if the are equal to each other or not, greater than one another, etc. Bitwise operators operates of individual bits of integers.

18. What do you mean by derived data type? What is local and global variable in C?

Answer:

Data types derived from basic data types, such as arrays, pointers, structures and unions are known as derived data type.

Local variable are the variables declared inside a function or block, accessible only within that function. Global variable are declared outside functions and are accessible throughout the program.

19. What is enum in C?

Answer:

It is a user-defined data type in C type in C that assigns names to a set of integer constants for better code readability.

20. What is an operator and operand?

Answer:

An operator refers the special symbols that are used in C to perform a certain action between two values or operand. Operand is the value on which that certain action is performed on.

21. What is the use of " #define " in C ?

Answer:

#define is used to create constants or macros to make the code easier to read and maintain.

22. What is an endless loop?

Answer:

An endless loop is a loop which keeps on running until the memory runs out because of logical error.

23. What are control statements?

Answer:

Control statements are used to check certain conditions defined by the programmer and execute tasks based on whether the conditions were met or not

24. What is || operator and how does it function in a program? Answer:

|| is a logical operator that performs the OR function. If any of the two condition is true, then || returns 1 or true, otherwise it returns 0 which represents false.

25. How to print 1 to 100 without using conditional operator or conditional statement?

Answer:

We can use loop to print 1 to 100 but technically they make the use of conditional operator, I can't think of any way of doing it apart from manually printing every number from 1 to 100.

- 26. Which bitwise operator is suitable for
 - a) turning on a particular bit in a number;
 - b) checking whether a particular bit is on or off?

Answer:

- a) &~
- b) ^

27. ERROR 404: QUESTION NOT FOUND

Answer:

Not able to answer this question.

28. Describe the difference between = and == symbols in C programming language?

Answer:

= used to assign a value to a variable, whereas == is used to check
if the two operands are equal to each other or not.

29. Which of the following operators is incorrect and why? (>=, <=, <>, ==)

Answer:

<> is incorrect because it is not defined in C.

30. Explain increment and decrements operators? What is the difference between the expression "++a" and "a++"?

Answer:

increment increases the variable's value by 1 where as decrements decrease by 1. ++a firsts increments the value, then uses the value, whereas a++ first uses the value and then increments it.

31. What does the format %10.2 mean when included in a printf statement?

Answer:

Do you mean %10.2f? It means that the the float will comprise of 10 digits out of which 2 would be after the decimal point.

32. What will be the outcome of the following conditional statement if the value of variable s is 10?

Answer:

Here we will solve the conditionals first, since they have higher precedence,

$$(s \ge 10) \&\& (s < 25) \&\& (s != 12)$$

Solving the conditions, we get,

Now, since all of the logical operator has the same precedence we will solve from left to right,

0 && 1

0

Therefore, the answer is 0.

33. Give the benefits of if else over conditional operators.

Answer:

If else is more readable and is very easy to expand into a nested if else, on the other hand, ternary operators are not very readable and gets overwhelmingly confusing to work in for nested statements.

34. Differentiate between if else and switch case with an example.

Answer:

if else checks for a condition to be true before executing the block underneath it, whereas in switch cases the compiler matches value to the appropriate case and executes the code in that code block.

```
//if else statement
if(a == 0)
    printf("ZERO");
else
    print("NON ZERO");

//switch case statement
switch(a){
    case 0:
        printf("ZERO");
        break;
    default: printf("NON ZERO");
}
```

35. Why most of the time it is necessary to use break statement ir switch case?

Answer:

Most of the time it is necessary to use break statement in switch case because if case is executed and it doesn't have a break statement then all the cases next are also executed by the c compiler, which we mostly don't want.

```
36. c = !a? 10 : 100; convert this statement into if and else form
Answer:
if (!a)
        c = 10;
else
        c = 100;
```

37. Can we use string as a case variable if yes give an example?

Answer:

No, we can't use string as a case variable in switch case.

38. Can we use continue instead of break to move program's execution to the start of switch? Explain.

Answer:

No we can't use 'continue' statement because it is used in loops to skip the current loop and not in switch case.

39. What value will be assigned to the variable X if a = 10, b = 20, c = 30, d = 40 for the expression X = a/b+c*d-c?

Answer:

Here / and * have higher precedence, so we get: x = (a/b) + (c*d) - c

So, we now get:

$$x = 0 + 1200 - 30$$

This gives us:

$$x = 1170$$

40. Is it possible to initialize a variable at the time it was declared? For initialization $a=2,\ c=1$ the value of a and c after this code will be

provided a = int, var1 = double, var2 = float

Answer:

Since, c = 1 which is true, a = 0 will be executed and the value of a would be assigned to c, giving the value of c to 0.

41. Define ternary operator in C. Which expression has to be present in the following?

```
exp1 ? exp2 : exp3;
```

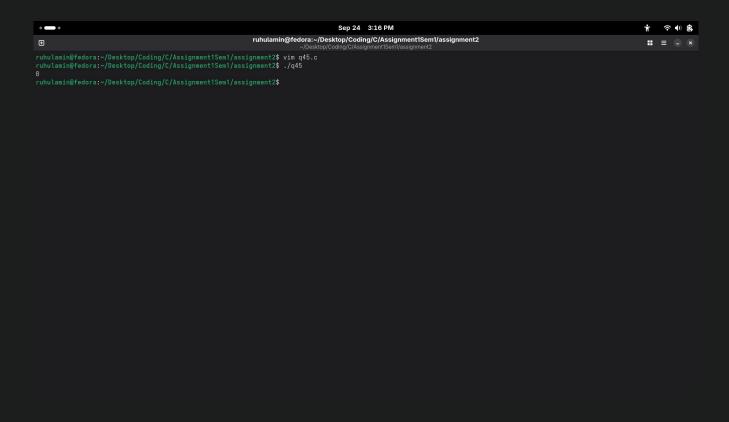
Answer:

Ternary operators are used to write if else statements in just a single line of code.

```
Answer:
Since -c = -1 which is not 0 it returns a value of 1. now c = c + a
will be executed, giving us the value of c = 2.
Answer:
x + y will give a float value but since it is been assigned to a int
data type so only the integer part would be assigned to y.
Answer:
ERROR: scanf("%d", %i);
FIX: scanf("%d", &i);
```

```
Answer:
ERROR: int main():
FIX: int main(){
ERROR: void main(){
FIX: removed
Answer:
ERROR: static int y = x;
FIX: int y = x;
```

Answer:



....

```
University of the state of the
```

Sep 24 3:17 PM

† ∻ ♦0 ®

```
Sep 24 :3:17 PM

□ ruhulamin@fedora-/Deakhg/Cadang/C/Assignment1Sem/assignment2 - vim q47.c

□ ruhulamin@fedora-/Deakhg/Cadang/C/Assignment2 - vim q47.c

□ ruhulamin@fedora-/Deakhg/Cadang/Cadangment2 - vim q47.c

□ ruhulamin@fedora-/Deakhg/Cadangment2 - vim q47.c

□ ruhulamin@fedora-/Deakhg/Cadan
```

Sep 24 3:18 PM

† **? ♦**) **©**

•

```
. .
                                                                                                                                               Sep 24 3:18 PM
                                                                                                                                                                                                                                                                                               ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/assignment2 — vim q48.c
☐ include <stdio.h>
Int main()
{
  int number=3,che_=3;
  if (number < 0||che_)
{
  printf("You entered %d.\n", number);
}
  printf("The if statement is easy.");
  return 0;
}
~
```

•	Sep 24 3:18 PM	π	৽ •) (
•	ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/assignment2 -/Desktop/Coding/C/Assignment1Sem1/assignment2	# :	≡ 6	
	ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/assignment2 ~/Desktop/Coding/C/Assignment1Sem1/assignment2 signment2\$ vim q48.c signment2\$./q48	::		

```
Sep 24 :3:18 PM

□ ruhulamin@fedora-/Deaktop/Codang/C/Assignment/2 - vim q49.c
□ ruhulamin@fedora-/Deaktop/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Codang/Coda
```

•	ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/assignment2 -//Desktop/Coding/C/Assignment1Sem1/assignment2	# ≡ - ×
ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/assi ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/assi 5 6 7ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1	ignment2\$ vim q49.c ignment2\$./q49	
3 0 /rundramingredora:~/besktop/coding/c/assignmentiSemi	7,855.ymmen.cz*	

Sep 24 3:18 PM

† **२♦**0 €

. . .



•	Sep 24 3:19 PM	† ∻ ቀን ®
•	ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/assignment2 ~/Desktop/Coding/C/Assignment1Sem1/assignment2	∷ ≡ • ×
ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/as ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/as Hiruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/	signment2\$ vim q50.c signment2\$./q50	

```
Tubulami@fedorat-yDesktop/Coding/c/Assignment3em/assignment2

Tubulami@fedorat-yDesktop/Coding/c/Assignment3em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/assignment2em/as
```

```
Sep 24 3:27 PM

in Unulaming fedora: -/Desktop/Coding/C/Assignment18 en/Assignment28 via c$2 c. ruhulaming fedora: -/Desktop/Coding/C/Assignment18 en/Assignment28 via c$2 c. ruhulaming fedora: -/Desktop/Coding/C/Assignment18 en/Assignment28 c. ruhulaming fedora: -/Desktop/Coding/C/Assignment28 via c$2 c. ruhulaming fedora: -/Desktop/Coding/C/Assignment28 via c. ruhulaming fed
```

```
nutuanin@fedora-/Desktop/Coding/C/Assignment1Sem/sssignment2 via Seminary (Seminary Coding/C/Assignment1Sem/sssignment2 via Seminary Coding/C/Assignment2 via Semi
```

```
Tuhulanin@fedora:-/Desktop/Coding/C/Assignment15em/ssignment28 vis g64.c ruhulanin@fedora:-/Desktop/Coding/C/Assignment15em/ssignment28 vis g64.c ruhulanin@fedora:-/Desktop/Coding/C/Assignment28 vis g64.c ruhulanin@fedo
```

```
| Sep 24 328 PW | Sep 24 328 P
```

```
Inhulaning fedora: -/Desktop/Coding/C/Assignment18em/ssignment28 vis qSs. e. ruhulaning fedora: -/Desktop/Coding/C/Assignment18em/ssignment28 vis qSs. e. ruhulaningfedora: -/Desktop/Coding/C/Assignment28 vis qSs. e. ruhulaningfedora: -/Desktop/Coding/C/A
```

```
Sep 24 328 PM

□ ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c

| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assignment12—vim q56.c
| ruhulamin@fedorar-(Desktop)Coding-(C)Assign
```

```
Sep 24 3:29 PM

□ ruhulamin@fedora:1-/Desktop/Coding/C/Assignment3em/assignment2

□ ruhulamin@fedora:4-/Desktop/Coding/C/Assignment3em/assignment2

- ruhulamin@fedora:4-/Desktop/Coding/C/Assignment3em/assignment2

- ruhulamin@fedora:4-/Desktop/Coding/C/Assignment3em/assignment2

- ruhulamin@fedora:4-/Desktop/Coding/C/Assignment3em/assignment2e

- ruhulamin@fedora:4-/Desktop/Coding/C/Assignment3em/assignment2e

- ruhulamin@fedora:4-/Desktop/Coding/C/Assignment3em/assignment2e

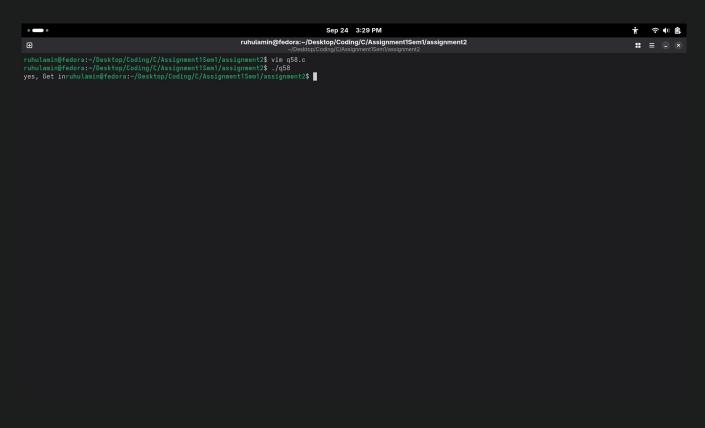
- ruhulamin@fedora:4-/Desktop/Coding/C/Assignment1sem/assignment2e

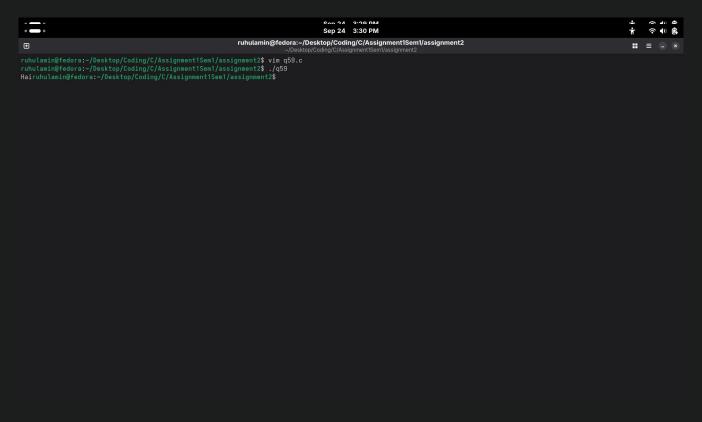
- ruhulamin@fedora:4-/Desktop/Coding/C/Assignment2e

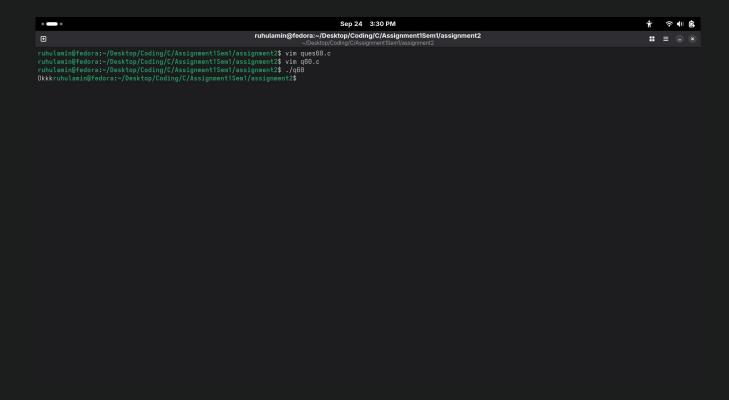
- ruhulamin@fe
```

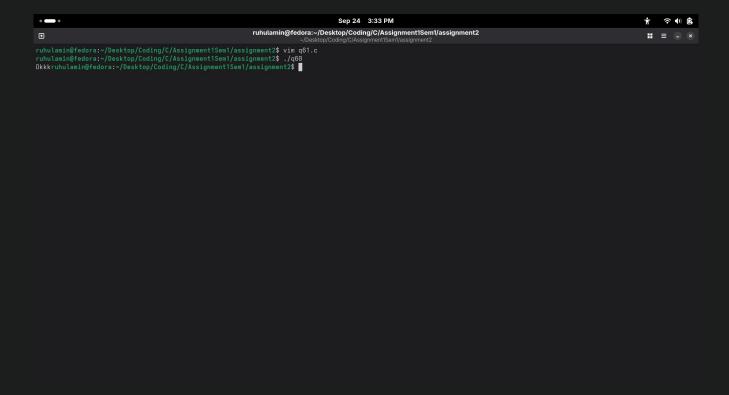
```
Industrial (Control Control Co
```

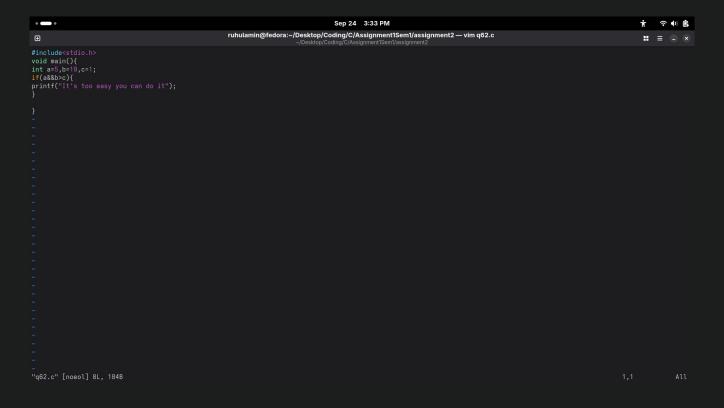
```
| Sep 24 329 PM | Sep 25 | Sep
```

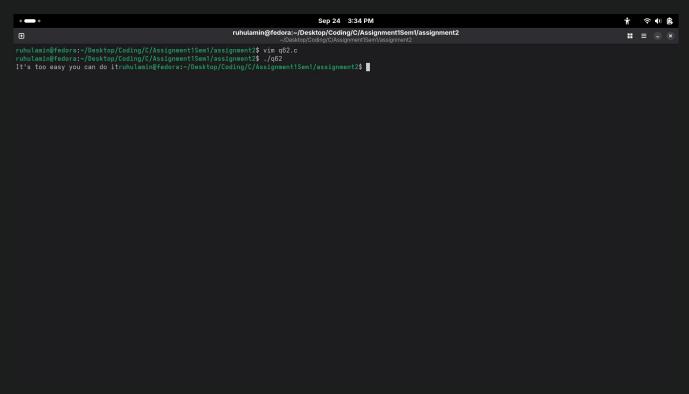












```
Sep 24 334 PN

□ ruhulamin@fedora-rDeaktop/Coding/C/Assignment12—vim q63.c
□ ruhulamin@fedora-rDeaktop/Coding/C/Assignment2-mim q63.c
□ ruhulamin@fedora-rDeaktop/C/Assignment2-mim q63.c
```

```
#Includestdio.h>

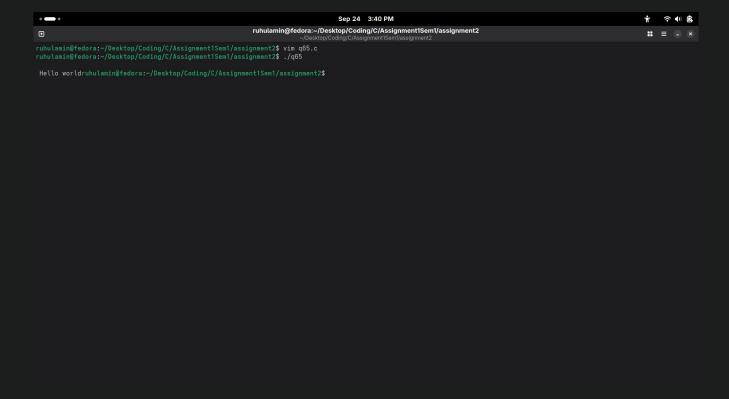
In the decision in the decis
```

```
Tululamin@fedora:-/Desktop/Coding/C/Assignment1Sem1/assignment2 via ng64.c

- Inhulamin@fedora:-/Desktop/Coding/C/Assignment1Sem1/assignment2 via ng64.c

- Inhulamin@fedora:-/Desktop/Coding/C/Assignment3 via ng64.c

- Inhulamin@fedora:-/Des
```



```
| Sep 24 341PM | Sep 26 341PM | Sep 26 341PM | Sep 26 34 341PM | Sep 26 36 341PM
```

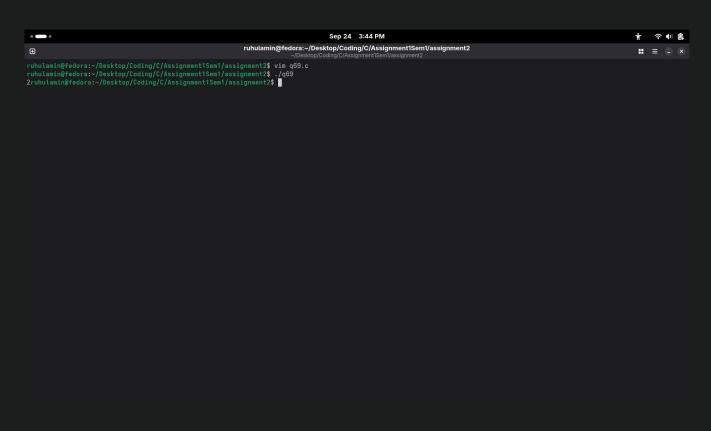
```
Sep 24 341PM

Chultumin@fedora:-/Desktop/Coding/CAssignment13 via d86.c rubultumin@fedora:-/Desktop/Coding/CAssignment13 via d86.c rubultumin@fedora:-/Desktop/Coding/C/Assignment13 via d
```

```
| Sep 24 341PM | Pick |
```

```
## Sep 24 3.44 PM

| Include cition | Position | Posit
```





ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/assignment2\$ vim q78.c ruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/assignment2\$./q70 TRUEruhulamin@fedora:~/Desktop/Coding/C/Assignment1Sem1/assignment2\$