



Light

Dark



GRASSROOTS 2.0

DAY 1

TOPICS COVERED:

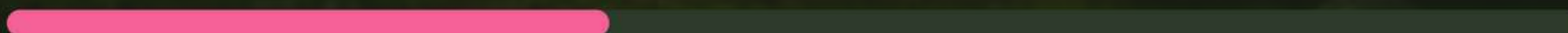
Installation setup

Variable, data types + input/output

Instructions & Operators

Conditional statements

Loop control statements



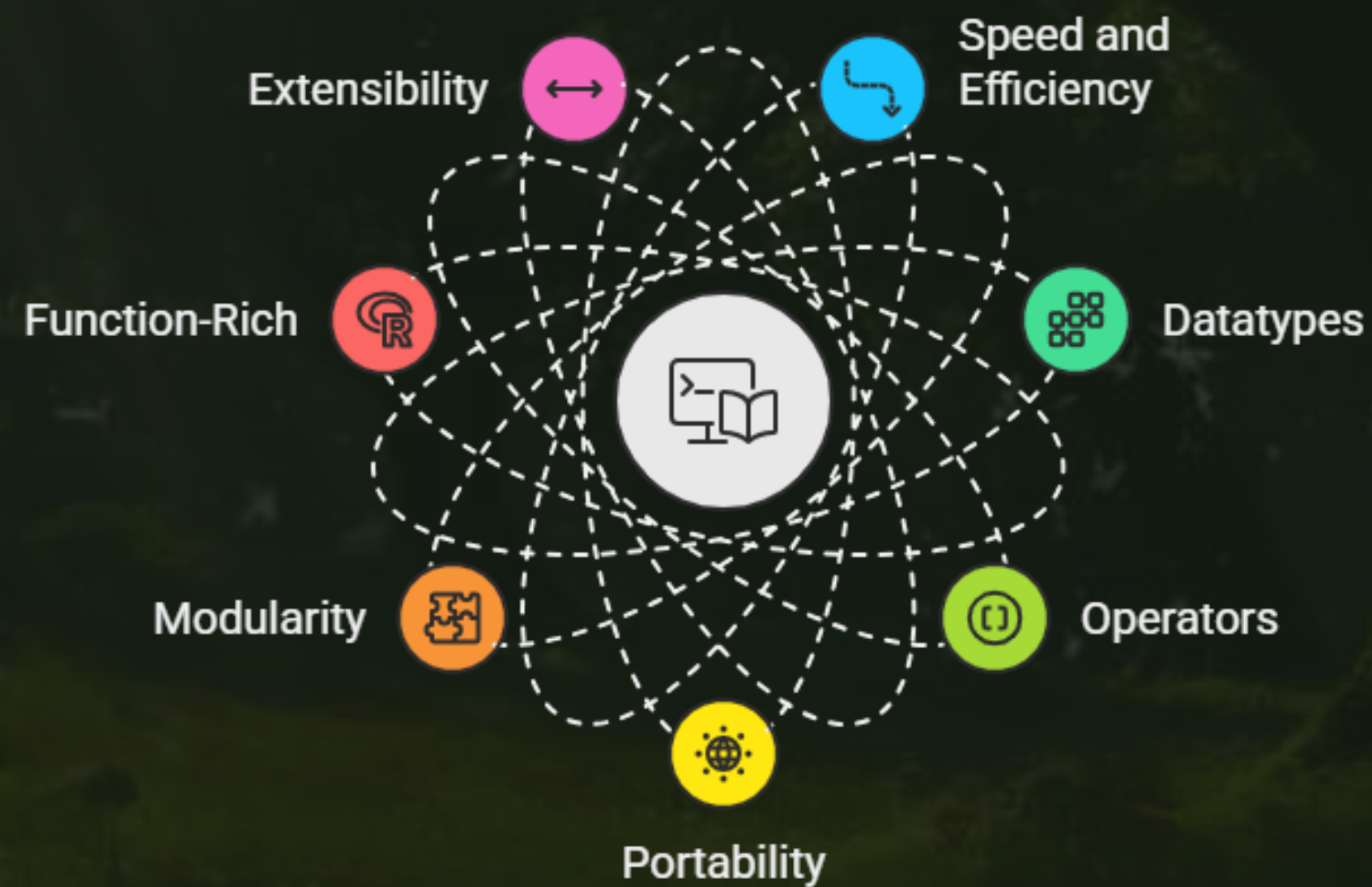


Light

Dark



Key Features of C Language





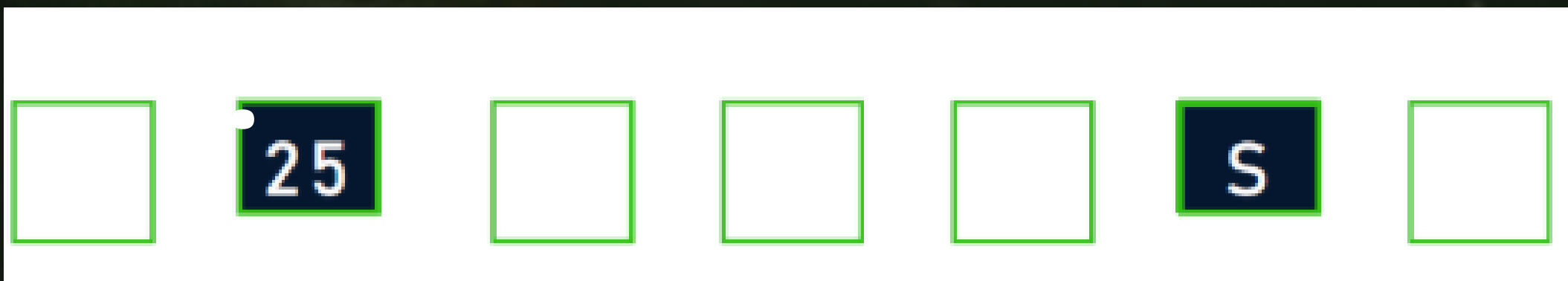
Light Dark



Variable

variables is the name of memory location which stores some data.

Memory





Light

Dark



Variable Rules

- a. Variables are case sensitive.
- b. First character is alphabet or '_'
- c. no commas / blank space
- d. No symbol other than '_'

A ≠ a

Variable Datatypes

Data type	Size in Bytes	Size in Bits
<code>char</code> / <code>signed char</code>	$1 = 2^0$	$8 = 2^3$
<code>unsigned char</code>	$1 = 2^0$	$8 = 2^3$
<code>short int</code> / <code>signed short int</code>	$2 = 2^1$	$16 = 2^4$
<code>unsigned short int</code>	$2 = 2^1$	$16 = 2^4$
<code>int</code> / <code>signed int</code>	$2 = 2^1$	$16 = 2^4$
<code>unsigned int</code>	$2 = 2^1$	$16 = 2^4$
<code>long int</code> / <code>signed long int</code>	$4 = 2^2$	$32 = 2^5$
<code>unsigned long int</code>	$4 = 2^2$	$32 = 2^5$
<code>float</code>	$4 = 2^2$	$32 = 2^5$
<code>double</code>	$8 = 2^3$	$64 = 2^6$
<code>long double</code>	10 (not exact power of 2)	80 (not exact power of 2)



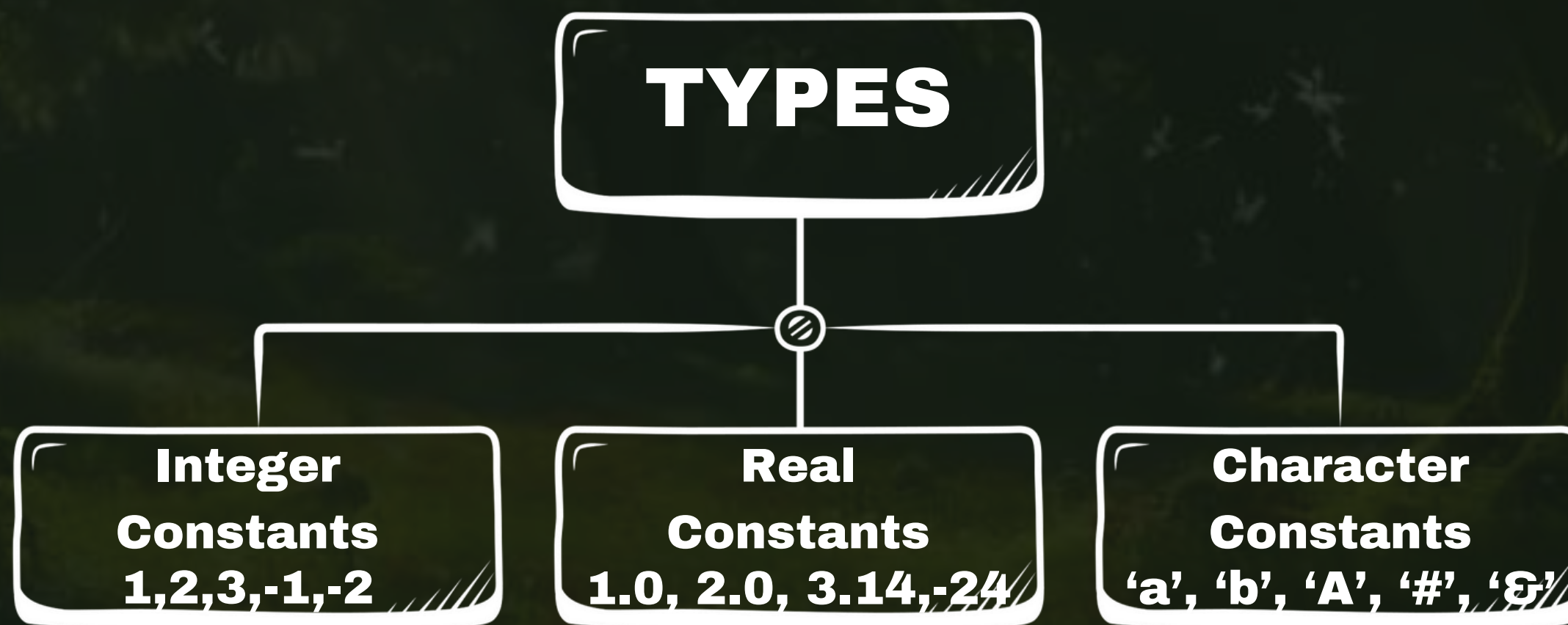
Light Dark



Constants

Values that don't change(fixed)

TYPES





Light

Dark



Keywords

Reserved words that have special meaning to the compiler



32 Keywords in C



Light

Dark



Keywords

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
continue	for	signed	void
do	if	static	while
default	goto	sizeof	volatile
const	float	short	unsigned



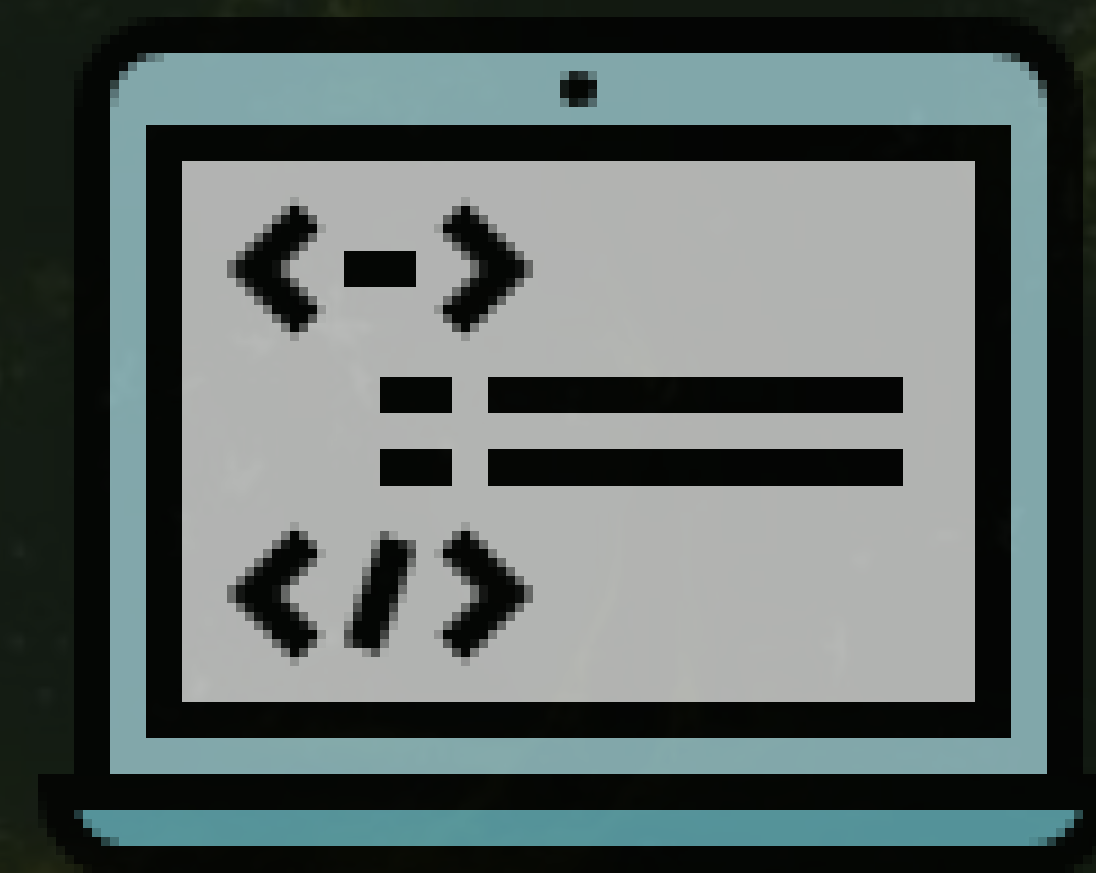
Light Dark



Program Structure

```
#include<stdio.h>

int main() {
    printf("Hello World")
    return 0;
}
```





Light Dark



Comments





Light

Dark



Output

```
printf(" Hello World");;
```

new line

```
printf(" kuch bhi \n");
```





Light Dark



Output

CASES

1. integers

```
printf(" age is %d ", age);
```

2. real numbers

```
printf(" value of pi is %f ", pi);
```

3. characters

```
printf(" star looks like this %c ", star);
```




Light

Dark



Input

```
scanf("%d", &age);
```



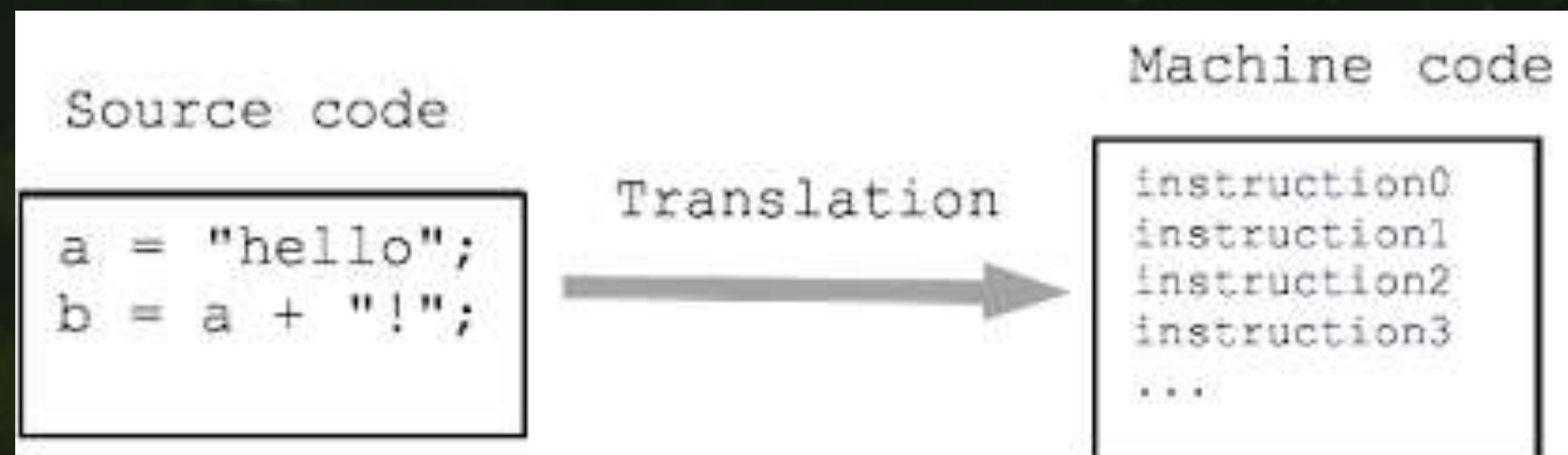


Light Dark



Compilation

A computer program that translates C codes into machine codes





Light

Dark



Problem 1

Write a program in C to store and display a user's name, age, and GPA. Use appropriate data types for each variable.

Input: Name (string), Age (integer), GPA (float).

Output: Display a message like: Hello [Name], you are [Age] years old and your GPA is [GPA].



Light Dark



Problem 2

Write code for a program that takes a temperature in Celsius as input (a float) and converts it to Fahrenheit using the formula: $F = (C * 9/5) + 32$. Display the result.

Input: Temperature in Celsius (float).

Output: Temperature in Fahrenheit (float).



Light Dark



Q. Write an algorithm and draw flowchart “**to convert temperature from Celsius to Fahrenheit**”.

Step 1 : Start.

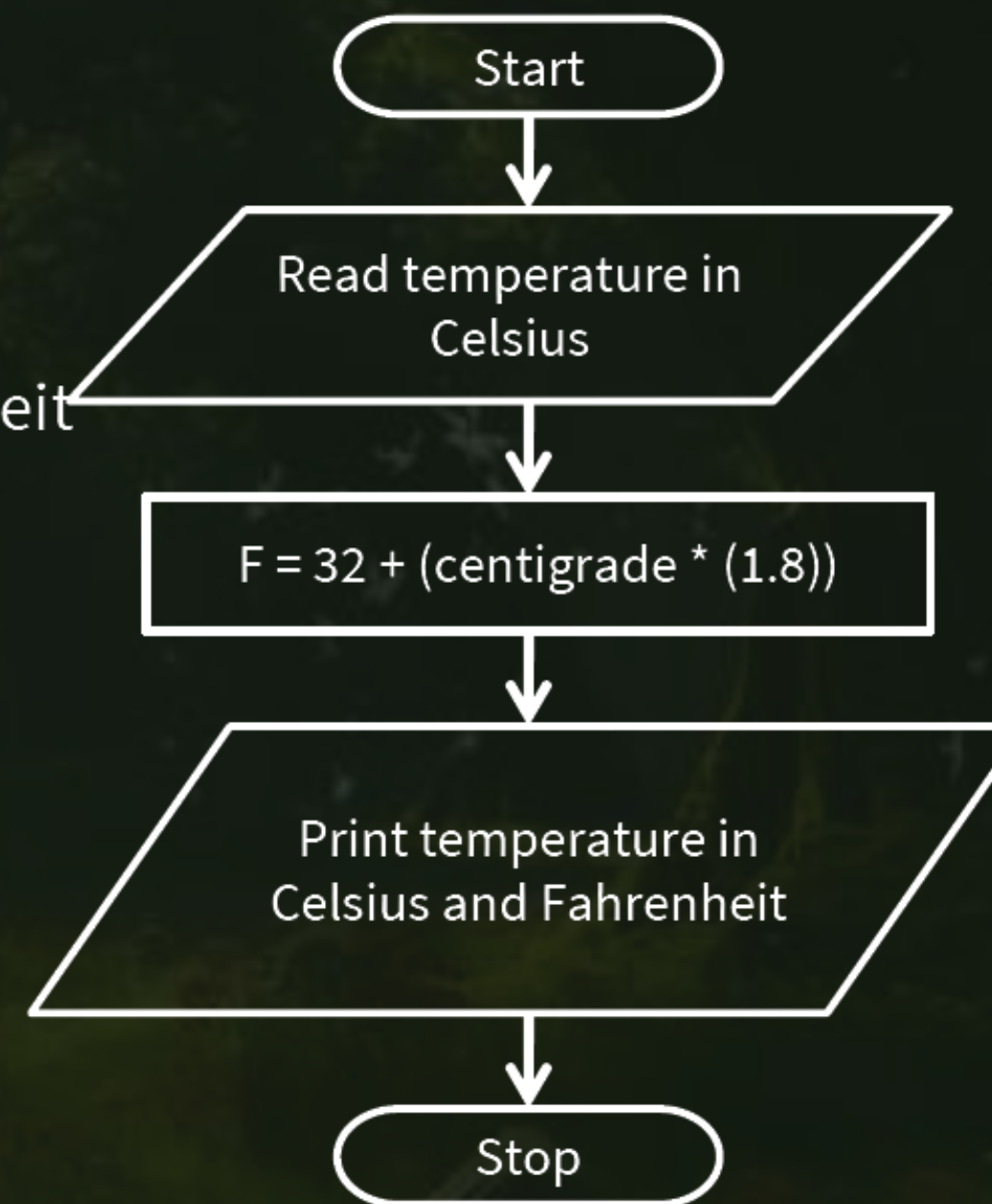
Step 2 : Read temperature in Celsius.

Step 3 : Calculate temperature in Fahrenheit

$$F = 32 + (\text{centigrade} * (1.8))$$

Step 4 : Display temperature in Celsius and Fahrenheit.

Step 5 : Stop.





Light Dark



Instructions

These are statements in a Program

Type Declaration
Instructions

Arithmetic
Instructions

Control
Instructions

Type Declaration Instructions → Declare var before using it

```
int a = 22;  
int b = a;  
int c = b + 1;  
int d = 1, e;  
  
int a,b,c;  
a = b = c = 1;
```

```
int a = 22;  
int b = a;  
int c = b + 2;  
int d = 2, e;  
  
int a,b,c = 1;
```




Light Dark



$a + b$

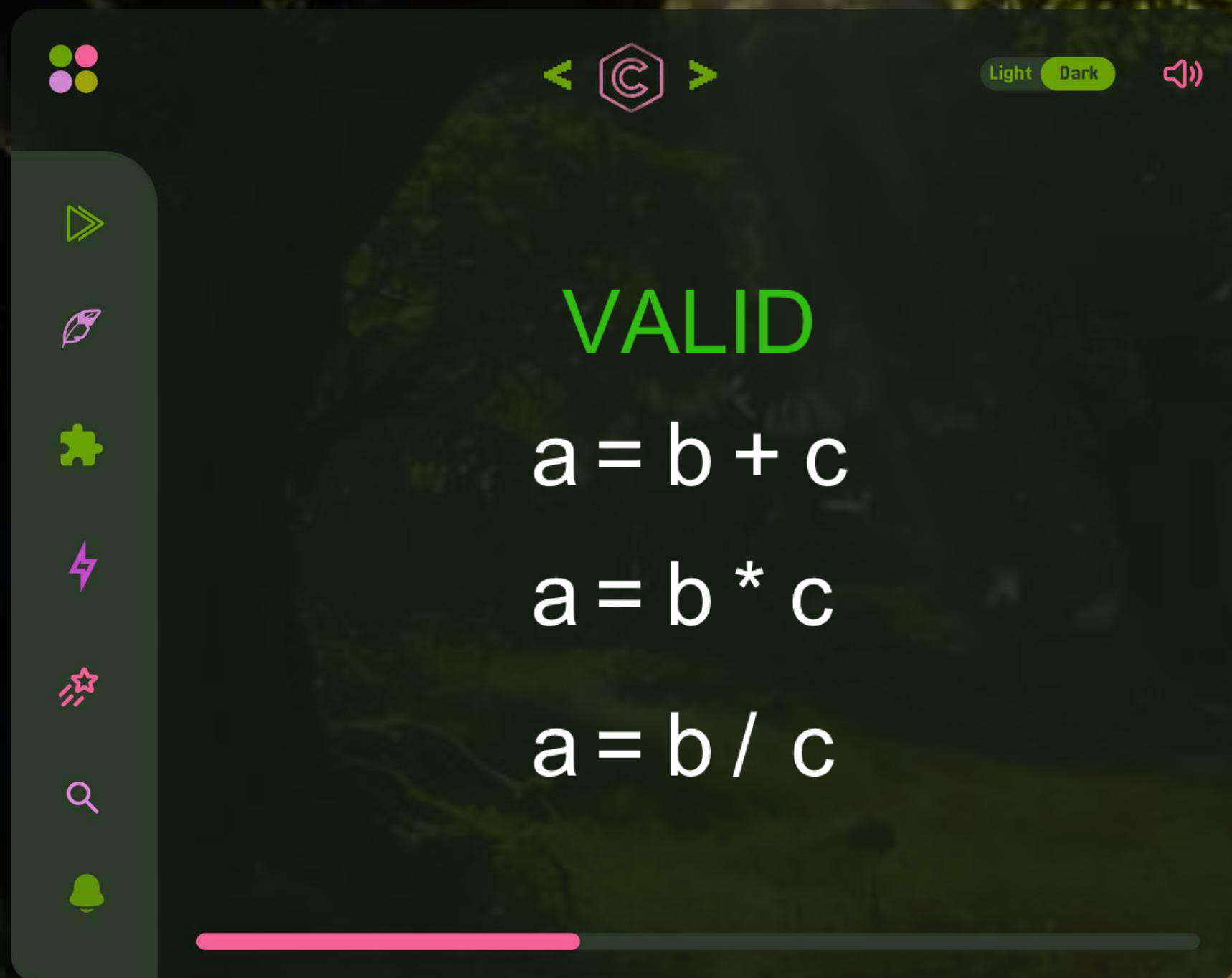
Operand 1

Operand 2

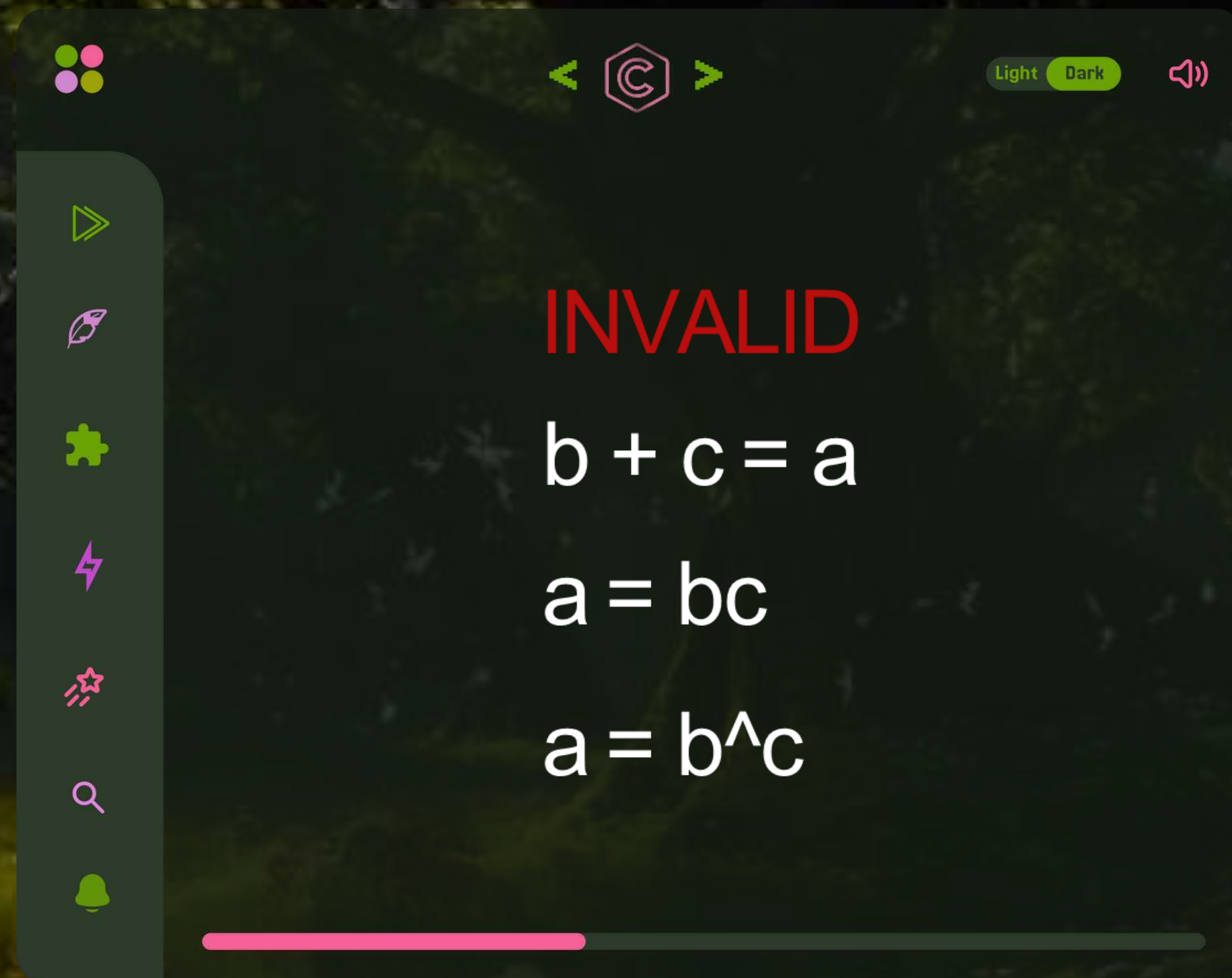
Operator

NOTE - single variable on the LHS

Arithmetic Instructions

A dark-themed interface for a code editor. At the top, there are icons for a menu (four colored dots), a C logo, and navigation arrows. Below these are 'Light' and 'Dark' theme toggles and a speaker icon. On the left, a vertical sidebar contains icons for a play button, a leaf, a puzzle piece, a lightning bolt, a star, a magnifying glass, and a bell. The main area displays the word 'VALID' in green, followed by three arithmetic instructions: $a = b + c$, $a = b * c$, and $a = b / c$. A pink progress bar is at the bottom.

VALID

$$a = b + c$$
$$a = b * c$$
$$a = b / c$$
A dark-themed interface for a code editor, identical in layout to the left one. It displays the word 'INVALID' in red, followed by three arithmetic instructions: $b + c = a$, $a = bc$, and $a = b^c$. A pink progress bar is at the bottom.

INVALID

$$b + c = a$$
$$a = bc$$
$$a = b^c$$

NOTE - $\text{pow}(x,y)$ for x to the power y



Light Dark



Arithmetic Instructions

★ Modular Operator %

Returns remainder for int

$$3 \% 2 = 1$$

$$-3 \% 2 = -1$$



Light Dark



Arithmetic Instructions

Type Conversion

int op int \longrightarrow int

int op float \longrightarrow float

float op float \longrightarrow float



Light Dark



Arithmetic Instructions

Operator Precedence

$*, /, \%$



$+, -$



$=$

$x = 4 + 9 * 10$

$x = 4 * 3 / 6 * 2$





Light Dark



Arithmetic Instructions

Associativity (for same precedence)

Left to Right

$$x = 4 * 3 / 6 * 2$$





Light Dark



Instructions

Control Instructions

Used to determine flow of program

- a. Sequence Control
- b. Decision Control
- c. Loop Control
- d. Case Control



Light Dark



Problem 3

To calculate the area of a circle given its radius. Use a constant value for π (3.14159) and the formula: $\text{Area} = \pi * \text{radius} * \text{radius}$.

Input: Radius of the circle (float)

Output: Area of the circle (float).



Light Dark



Problem 4

Write a program that checks whether a given integer is even or odd. Use the modulus operator (%) to determine divisibility.

Input: An integer (int).

Output: A message indicating whether the number is even or odd.



Light Dark



Operators

a. Arithmetic Operators

b. Relational Operators

c. Logical Operators

d. Bitwise Operators

e. Assignment Operators

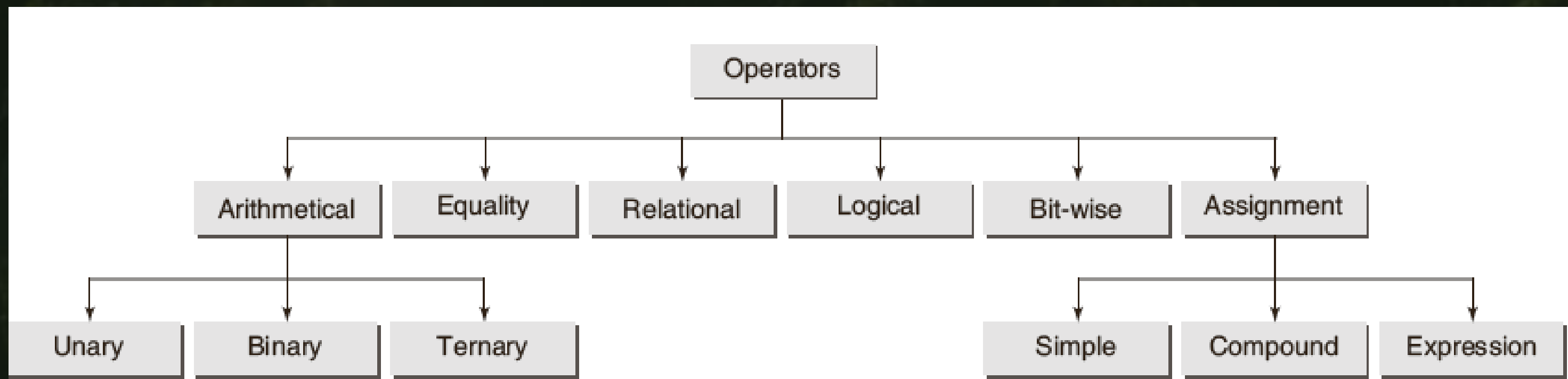
f. Ternary Operator



Light Dark



Operators





Light Dark



Relational Operator

==

<, <=

>, >=

!=



Light Dark



Logical Operator

&& AND

|| OR

! NOT



Operator precedence

Priority	Operator
1	!
2	*, /, %
3	+, -
4	<, <=, >, >=
5	==, !=
6	&&
7	
8	=



Light Dark



Assignment Operator

=

*=

+=

/=

-=

%=



Light Dark



Problem 5

Write a program that takes two integers as input from the user and performs the following arithmetic operations:

- Addition
- Subtraction
- Multiplication
- Division
- Modulus (remainder)

Input: Two integers: num1 and num2

Output: Print the result of each operation in the following format:

- num1 + num2 = [result]
- num1 - num2 = [result]
- num1 * num2 = [result]
- num1 / num2 = [result]
- num1 % num2 = [result]



Light Dark



Problem 6

Write a program to swap the values of two variables using the **bitwise XOR operator** without using a third variable.

Input: Two integers: a and b

Output: Print the values of a and b before and after swapping in the following format:

- Before swapping: a = [value], b = [value]
- After swapping: a = [new value], b = [new value]



Light Dark



Conditional\Selection Statement

Conditional selection statements, which run different statements for different data values. The conditional selection statements are IF and CASE .

The conditional operator has three expressions.

➤ It has the general form

expression1 ? expression2 : expression3



(Condition) ? (True statement) : (False statement)

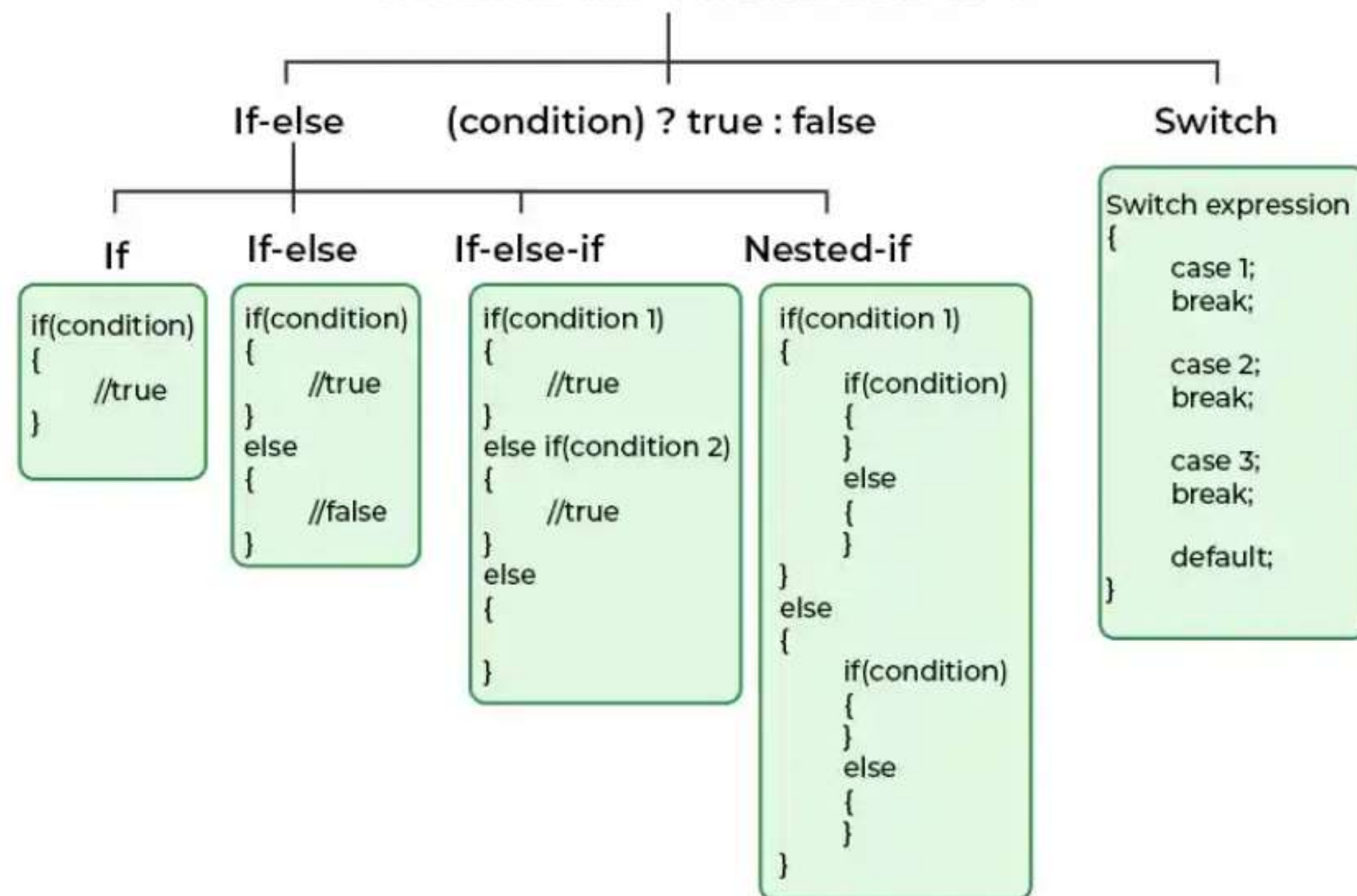


Light

Dark



Conditional Statements in C





Light Dark



Problem 7

Write a problem to compare 3 variables to find largest number using if-else rule

Input: 3 variables assigned with a number each

Output: Comparing which variable is largest among the 3



Light Dark



Two-way decisions using **if-else** statement



if-else statement is a conditional expression that executes a specific block of code if a condition is true, or another block of code if the condition is false



Syntax :



```
if (condition) {  
    // code executed when the condition is true  
}  
else {  
    // code executed when the condition is false  
}
```





Light Dark



Multi-way decisions using **if-else-if** ladder

if-else-if ladder, is a programming construct that evaluates multiple conditions and executes different statements based on those conditions

Syntax:

```
if(condition) {  
    // any if-else ladder starts with an if statement only  
}  
else if(condition) {  
    // this else if will be executed when condition in if is false and  
    // the condition of this else if is true  
}  
.... // once if-else ladder can have multiple else if  
else { // at the end we put else  
}
```




Light Dark



Switch Statement

A switch-case statement is a programming construct that executes different code blocks based on specified conditions. It's similar to a series of nested if/else statements, and includes a default block to run when no conditions are met.

Syntax :

```
switch(expression)
{
    case value1: statement_1;
                break;
    case value2: statement_2;
                break;
    .....
    case value_n: statement_n;
                break;
    default: default_statement;
}
```




Light Dark



Problem 8

Write a program to display grade of a student base on marks using switch case : Grade A : 80-100 , Grade B: 60-79, Grade C : 40-59 Fail: 0-39

Input: Insert the grade of student

Output: Student grade





Light

Dark



Loop Statements

The switch case in C executes one statement from multiple ones.





Light

Dark



Loops

Entry Controlled

Exit Controlled

for

while

do-while

```
for( initialization ; condition; updation)
{
}
```

```
while( condition )
{
}
```

```
do
{
}while( condition )
```




Light Dark



while Loop

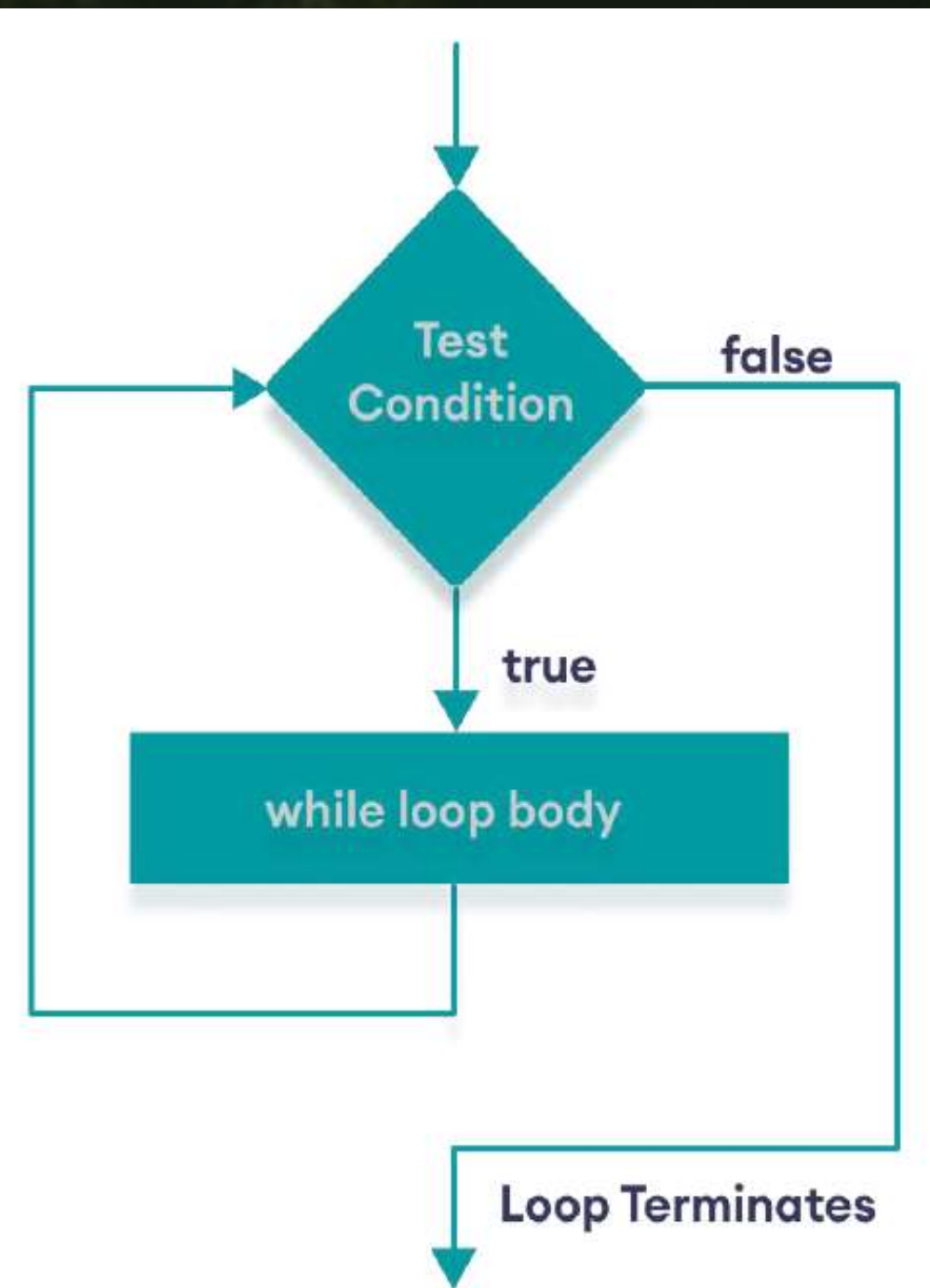
A while loop is a control flow statement in programming languages that repeats a block of code based on a condition

Syntax :

```
while (testExpression) {  
    // the body of the loop  
}
```



while Loop





Light Dark



Problem Statement 9

Write a program to display table of any number entered by user.

Input: Integer for which table is to be generated

Output: Multiplication table for the given number



Light Dark



for Loop

A for loop is a programming construct that repeatedly executes a block of code until a condition is met

Syntax :

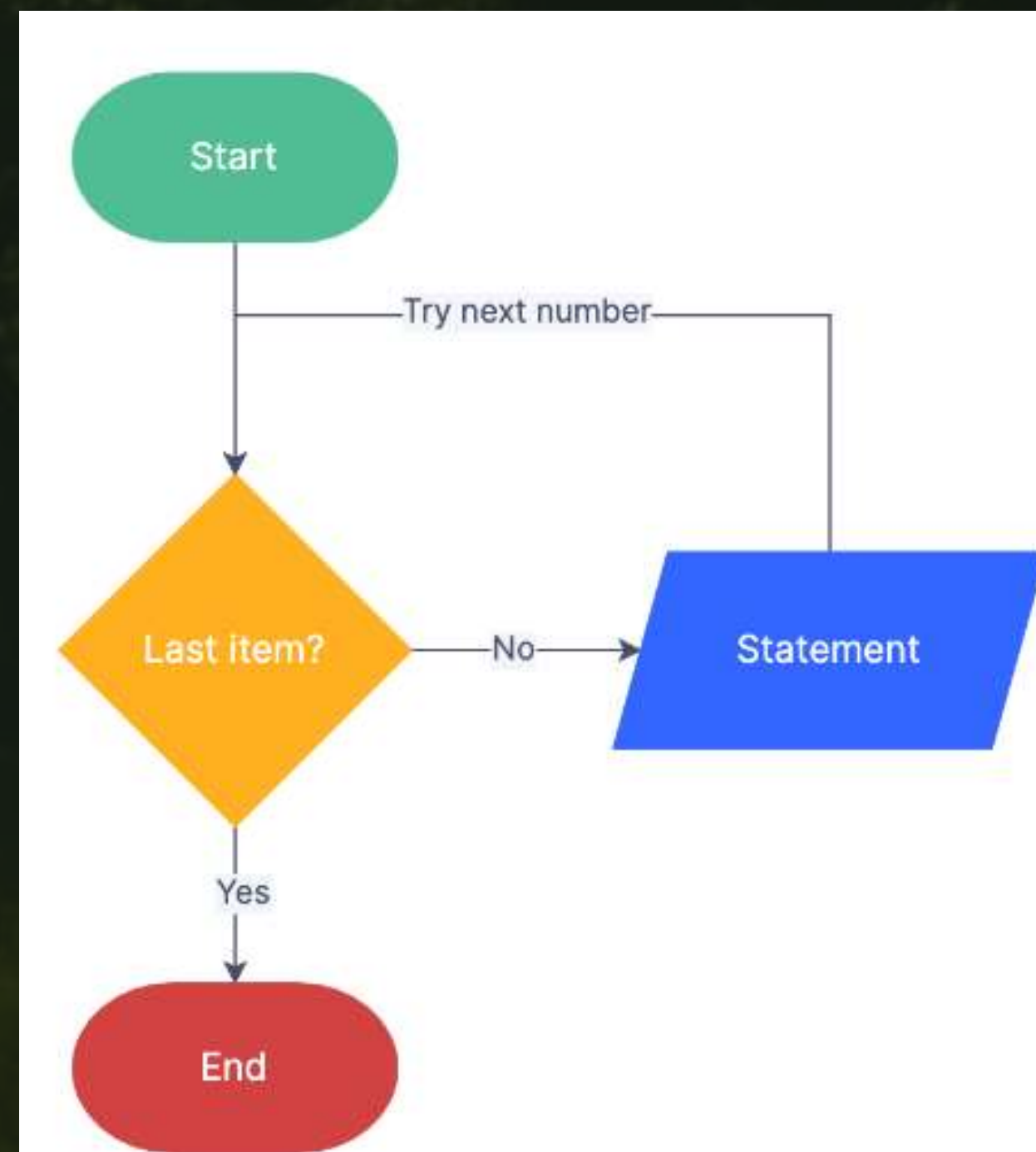
```
for (declarationStatement; testExpression;  
    updateStatement)  
{  
    // statements inside the body of loop  
}
```




Light Dark



for Loop



Problem Statement 10

Write a program to print sum of 1 to n numbers

Input: Enter the limit (any integer)

Output: Summation of 1 to limit

Iteration	Variable	$i \leq 5$	Action
1st	$i=1$	true	1 is added in sum i.e. $0+1=1$ & "i" is increased by 1 i.e. $i=2$.
2nd	$i=2$	true	2 is added in sum i.e. $1+2=3$ & "i" is increased by 1 i.e. $i=3$.
3rd	$i=3$	true	3 is added in sum i.e. $3+3=6$ & "i" is increased by 1 i.e. $i=4$.
4th	$i=4$	true	4 is added in sum i.e. $6+4=10$ & "i" is increased by 1 i.e. $i=5$.
5th	$i=5$	true	5 is added in sum i.e. $10+5=15$ & "i" is increased by 1 i.e. $i=6$.
6th	$i=6$	false	For loop will be terminated



Light Dark



do while loop

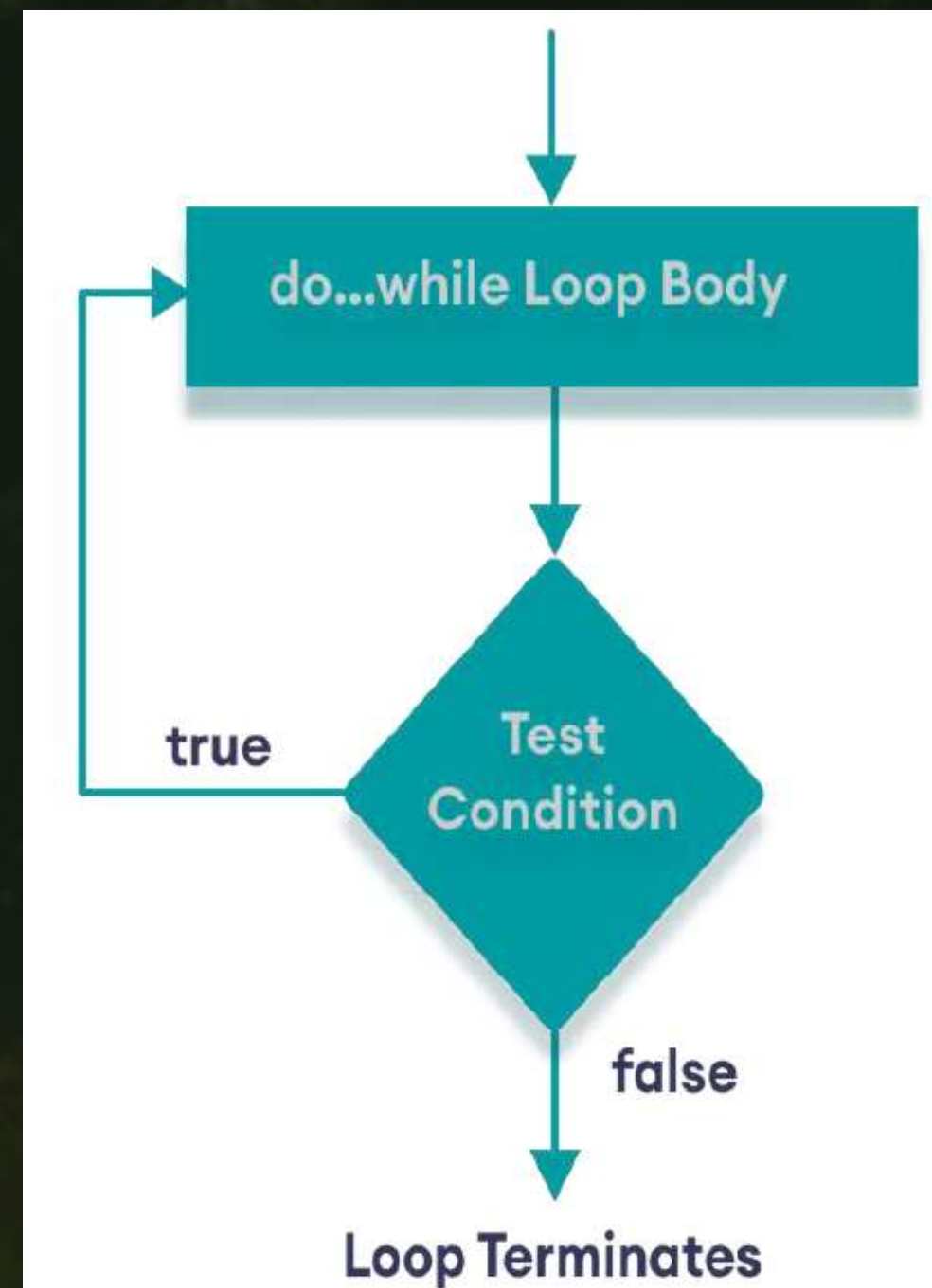
A do while loop is a control flow statement in programming that executes a block of code at least once, and then repeats the block based on a condition

Syntax :

```
do {  
    // the body of the  
    loop  
}
```

~~while (testExpression);~~

do while loop





Light Dark



Problem Statement 11

Write a program to display table of any number entered by user.

Input: Integer for which table is to be generated

Output: Multiplication table for the given number





Light Dark



while	do-while
Syntax: while(test expression) { statement }	Syntax: do { statement } while(test expression);
Condition is checked first and then statement is executed. Hence it is entry controlled loop.	Statement is executed at least once thereafter condition is checked. Hence it is exit controlled loop.
This is pre test loop.	This is post test loop.
If condition is not true, for first iteration the loop will never get executed.	Though condition is not true for first iteration , the loop will be executed once.
No semicolon is given after while statement.	While statement ends with semicolon.



Light

Dark



Thank You

