AGE-GROUP SURVEY ANALYZER

Presented by:- Faiz Ahmad

Guided by :- Naina Ma’am

RUNGTA
INTERNATIONAL SKILLS
UNIVERSITY

INTRODUCTION

- The Age-Group Survey Analyzer is a mini project develop using the C programming language.
- The project is designed to collect age data from multiple users efficiently.
- It classifies users into different age groups based on predefined age ranges.
- The project demonstrates the practical use of conditional statements and loops.
- It helps beginners understand how programming concepts solve real-world problems.

OBJECTIVE

- TO COLLECT AGE DATA FROM USERS IN A SYSTEMATIC MANNER
- TO CLASSIFY INDIVIDUALS INTO PREDEFINED AGE GROUPS
- TO ANALYZE DEMOGRAPHIC DISTRIBUTION EFFICIENTLY
- TO APPLY FUNDAMENTAL C PROGRAMMING CONCEPTS TO A REAL-WORLD PROBLEM

PROBLEM STATEMENT

1. Manual age analysis is time-consuming
2. High chances of human error in calculations
3. Need for an automated age classification system
4. Simple and efficient solution using C programming

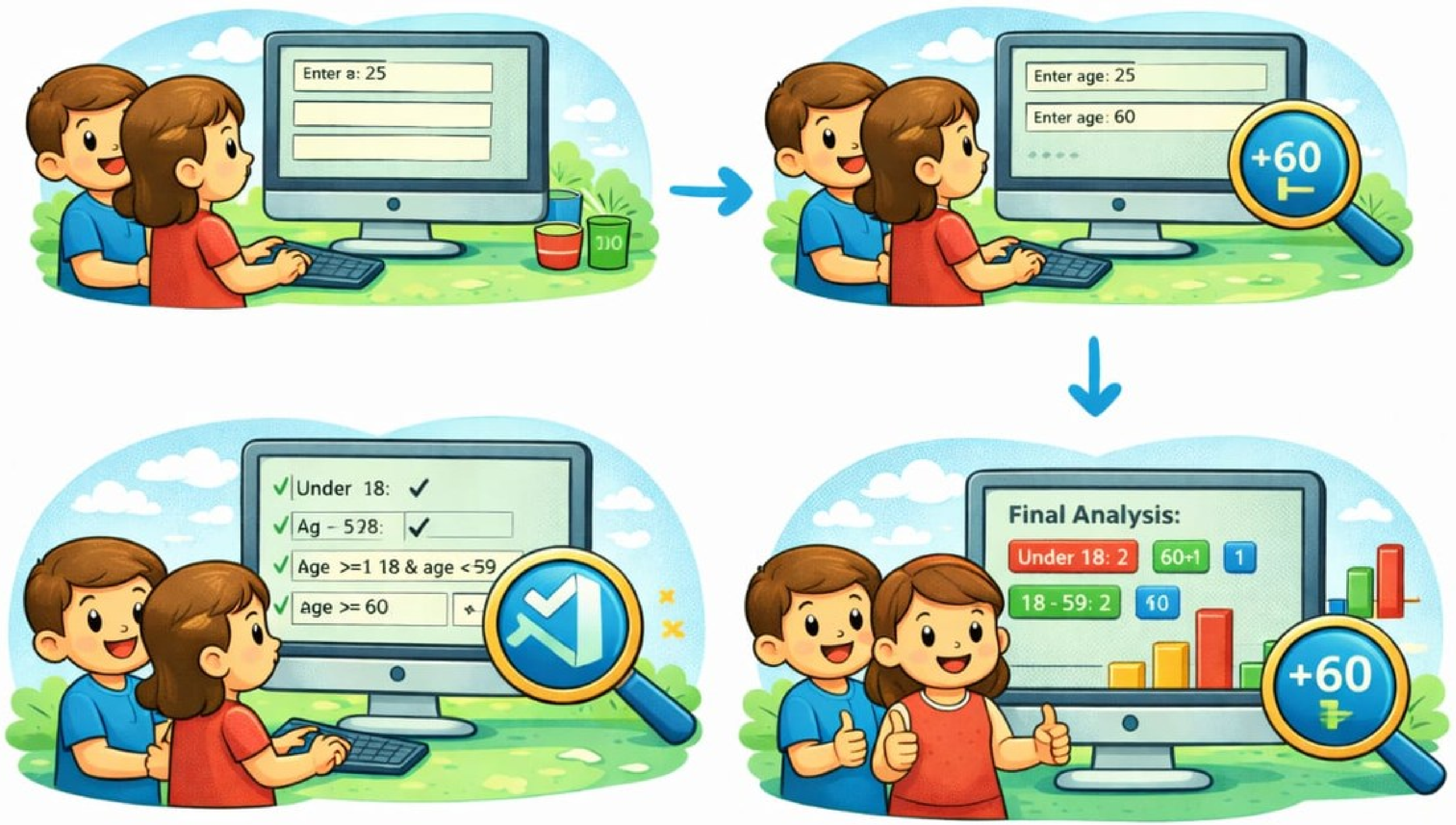
TOOLS AND TECHNOLOGIES

- **Programming Language:** C
- **Development Environment:** Visual Studio Code
- **Operating System:** Windows / Linux
- **Concepts Used:**

1. Conditional statements
2. Looping structures
3. Counters and variables
4. Input/output operations

PROJECT WORKING

1. User enters number of individuals
2. Program accepts age of each use
3. Conditions check age range
4. Age group counters are updated
5. Final analysis is displayed



AGE GROUP CLASSIFICATION

0 – 12 years → Child

13 – 19 years → Teenager

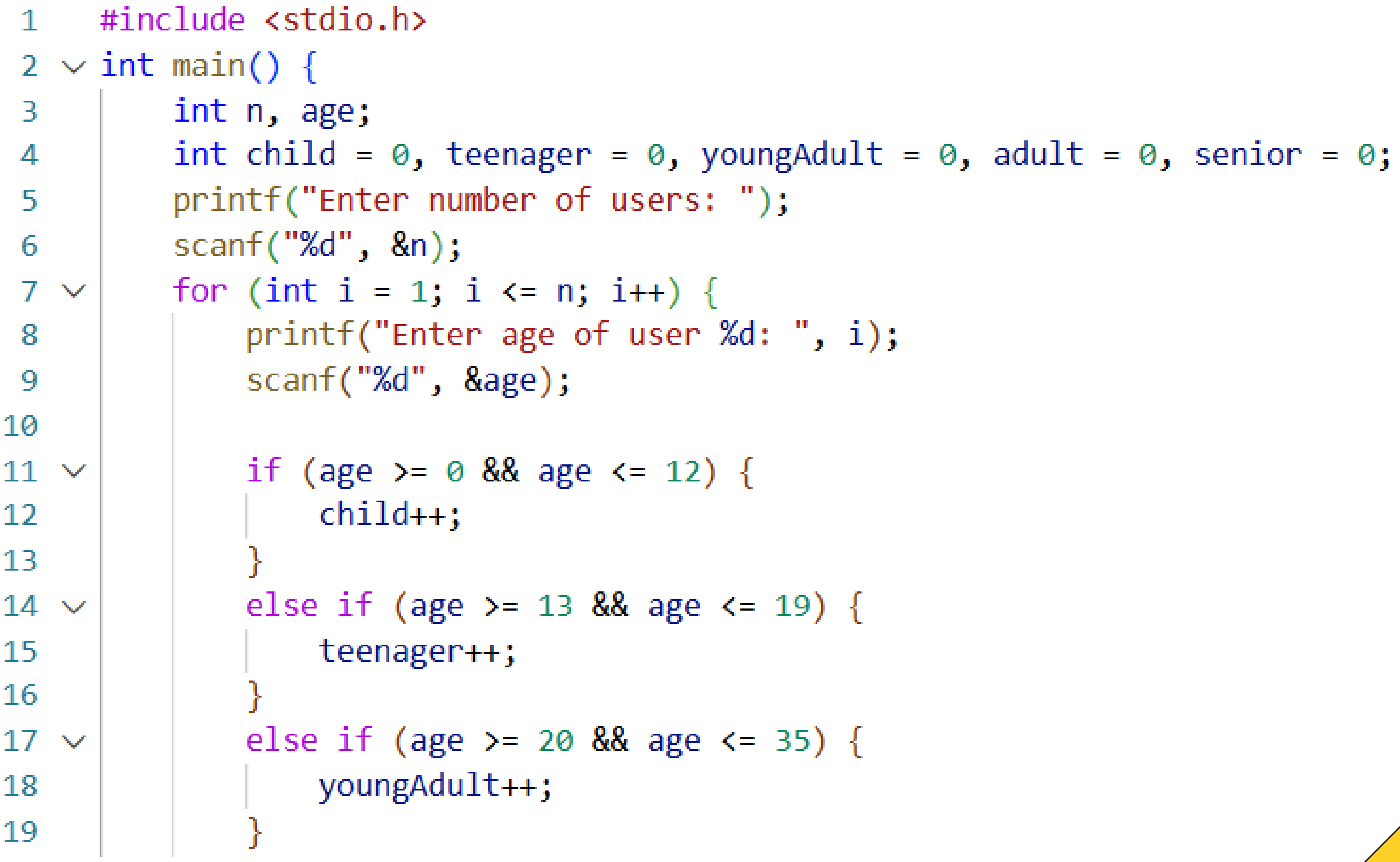
20 – 35 years → Young Adult

36 – 59 years → Adult

60+ years → Senior Citizen

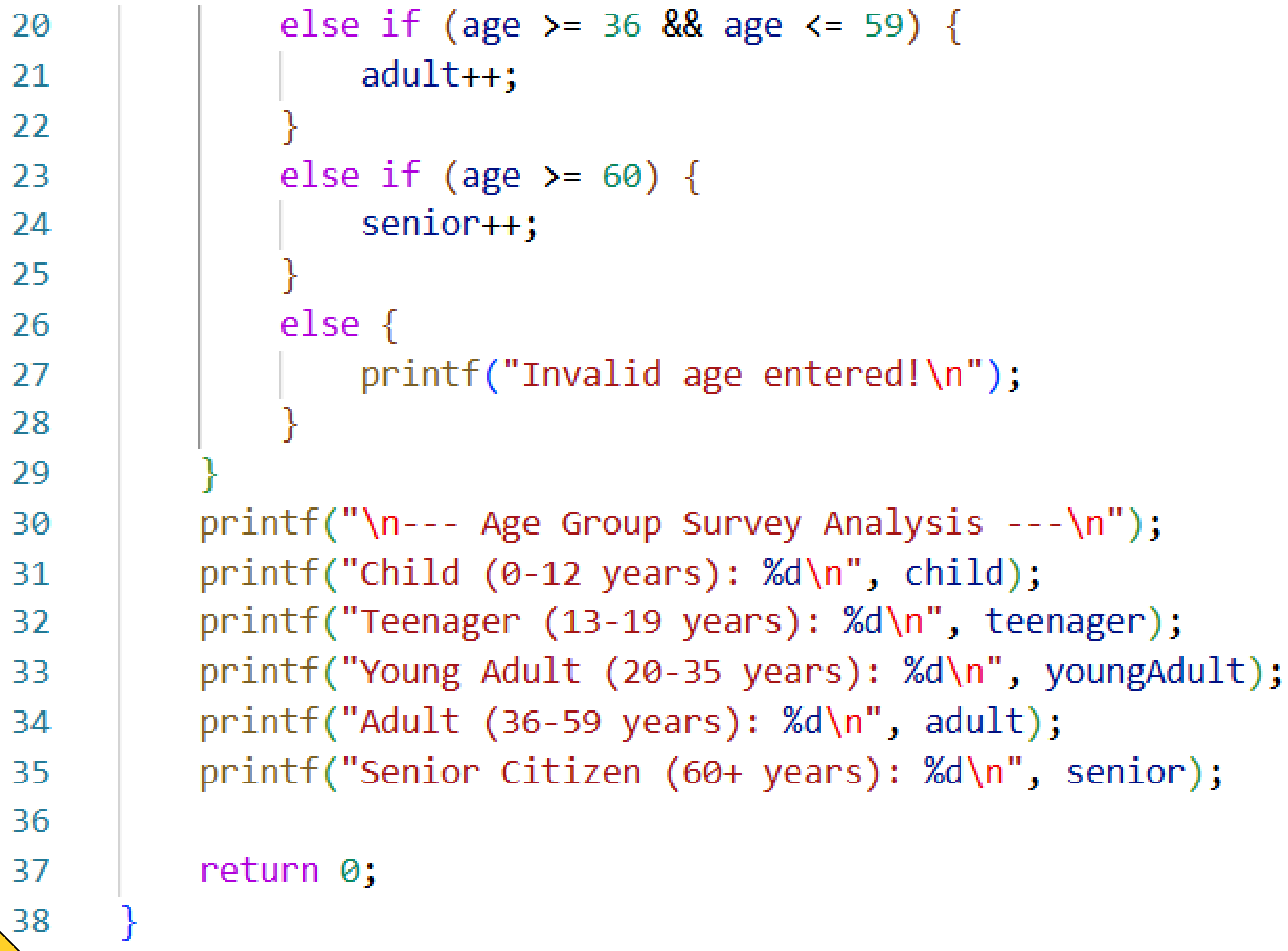
0 - 12 years
13 - 19 years
20 - 35 years
36 - 59 years
60 + years
Child
Teenager
Young Adult
Adult
Senior Citizen
Age Group Classification

SOURCE CODE (C PROGRAM – PART 1)



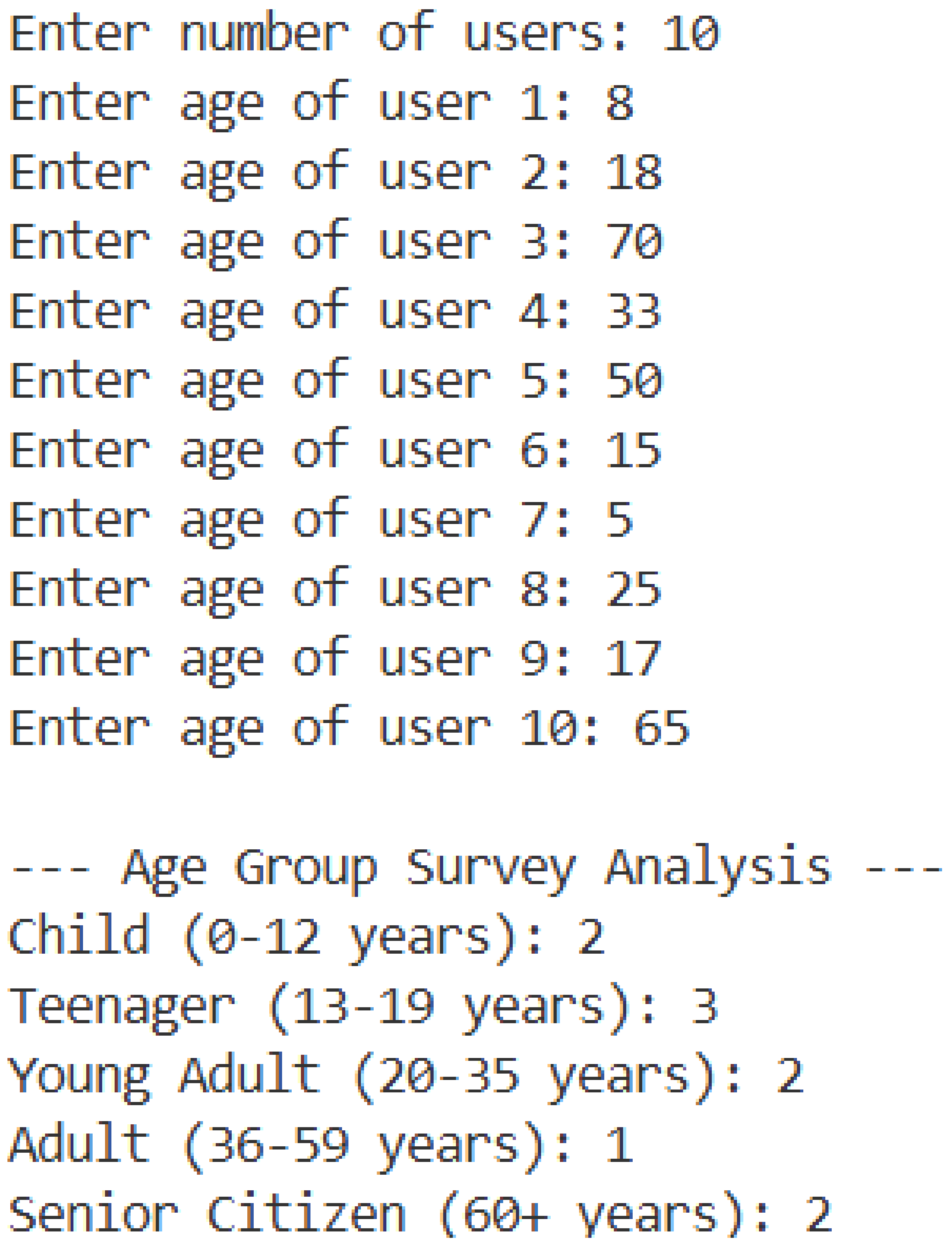
```c
#include <stdio.h>
int main() {
    int n, age;
    int child = 0, teenager = 0, youngAdult = 0, adult = 0, senior = 0;
    printf("Enter number of users: ");
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) {
        printf("Enter age of user %d: ", i);
        scanf("%d", &age);

        if (age >= 0 && age <= 12) {
            child++;
        }
        else if (age >= 13 && age <= 19) {
            teenager++;
        }
        else if (age >= 20 && age <= 35) {
            youngAdult++;
        }
```

SOURCE CODE (C PROGRAM – PART 2)



```c
        else if (age >= 36 && age <= 59) {
            adult++;
        }
        else if (age >= 60) {
            senior++;
        }
        else {
            printf("Invalid age entered!\n");
        }
    }
    printf("\n--- Age Group Survey Analysis ---\n");
    printf("Child (0-12 years): %d\n", child);
    printf("Teenager (13-19 years): %d\n", teenager);
    printf("Young Adult (20-35 years): %d\n", youngAdult);
    printf("Adult (36-59 years): %d\n", adult);
    printf("Senior Citizen (60+ years): %d\n", senior);

    return 0;
}
```

PROGRAM OUTPUT



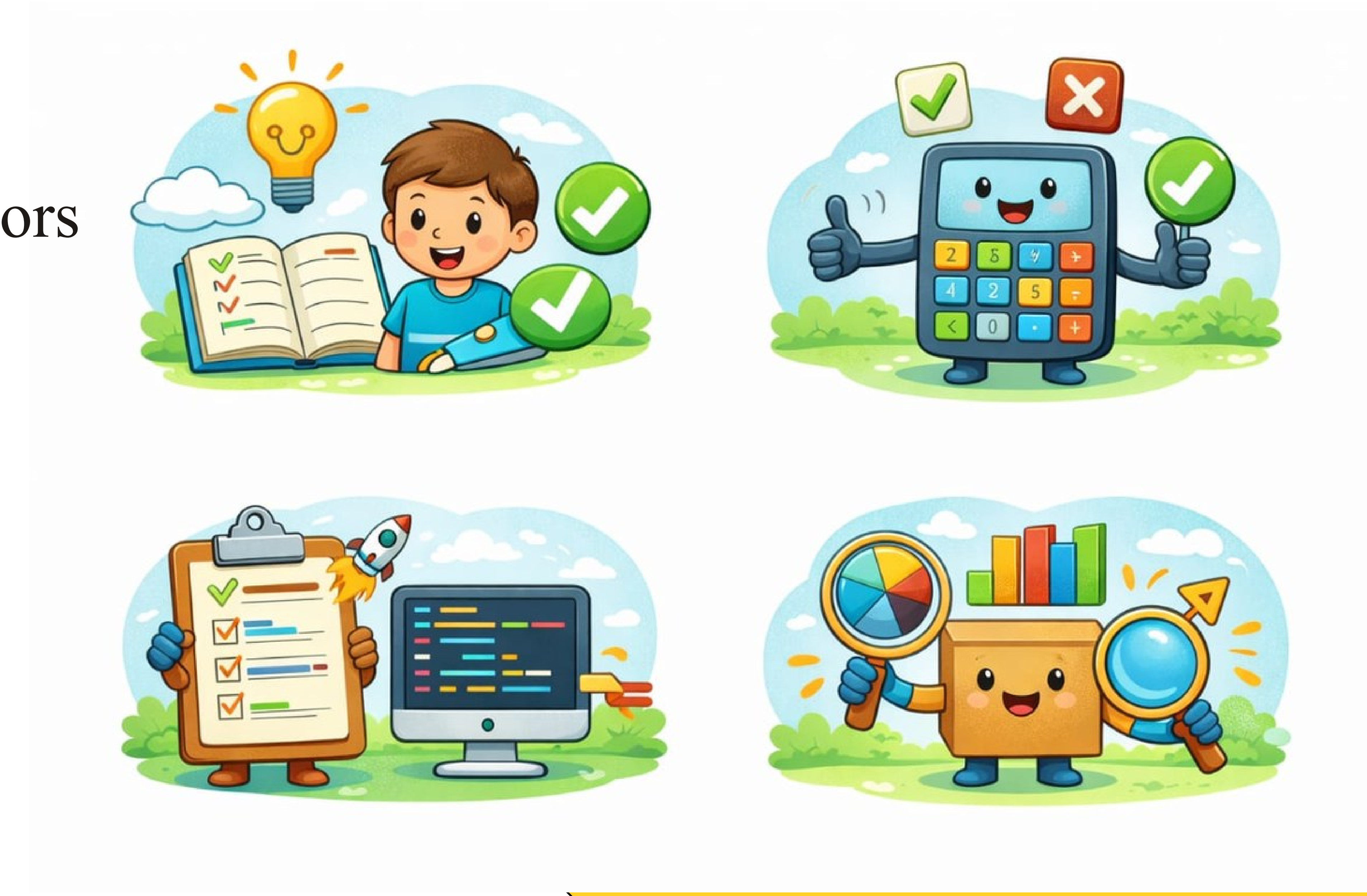
```
Enter number of users: 10
Enter age of user 1: 8
Enter age of user 2: 18
Enter age of user 3: 70
Enter age of user 4: 33
Enter age of user 5: 50
Enter age of user 6: 15
Enter age of user 7: 5
Enter age of user 8: 25
Enter age of user 9: 17
Enter age of user 10: 65

--- Age Group Survey Analysis ---
Child (0-12 years): 2
Teenager (13-19 years): 3
Young Adult (20-35 years): 2
Adult (36-59 years): 1
Senior Citizen (60+ years): 2
```

APPLICATIONS

- Population and demographic surveys
- Educational institutions
- Market research studies
- Learning basic programming concepts

ADVANTAGES

- Simple and easy to understand
- Reduces manual calculation errors
- Beginner-friendly project
- Efficient for small datasets



FUTURE SCOPES

- Add file handling for data storage
- Show percentage and graphical analysis
- Convert into menu-driven program
- Develop GUI-based application

CONCLUSION

- Successfully analyzes age data
- Demonstrates practical use of C programming
- Improves logical thinking and coding skills
- Suitable for beginner-level projects

THANK YOU