New Task:
Build a Rust-based console app that:

- Stores student's name, total marks & number of subjects
- Calculates average using a custom function
- Assigns grade:
  - A: 90+
  - B: 75-89
  - C: 60-74
  - D: Below 60
- Prints a clean report card in a PDF

Cargo.toml:

```
[package]
name = "rust_report_card"
version = "0.1.0"
edition = "2021"

[dependencies]
printpdf = "0.3.4"
chrono = "0.4"
```

src/main.rs:

```
use std::io;
use printpdf::*;
use std::fs::File;
use std::io::BufWriter;
use chrono::Local;

fn calculate_average(total_marks: f64, num_subjects: u32) -> f64 {
    total_marks / num_subjects as f64
}

fn assign_grade(avg: f64) -> char {
    if avg >= 90.0 { 'A' } else if avg >= 75.0 { 'B' }
    else if avg >= 60.0 { 'C' } else { 'D' }
}

fn generate_pdf(name: &str, total: f64, subjects: u32, average: f64, grade: char) {
    let (doc, page1, layer1) = PdfDocument::new("Report Card", Mm(210.0), Mm(297.0), "Layer 1
    let layer = doc.get_page(page1).get_layer(layer1);
    let font = doc.add_builtin_font(BuiltinFont::Helvetica).unwrap();

    let date_str = Local::now().format("%Y-%m-%d").to_string();
    let lines = vec![
        "------------------------------".to_string(),
        "       Student Report Card".to_string(),
        "------------------------------".to_string(),
        format!("Name of Student   : {}", name),
        format!("Total Marks       : {}", total),
        format!("Number of Subjects: {}", subjects),
        format!("Average Marks     : {:.2}", average),
        format!("Grade Assigned    : {}", grade),
        format!("Generated On      : {}", date_str),
        "".to_string(),
        "Grade Criteria:".to_string(),
        "  A: 90 and above".to_string(),
        "  B: 75 to 89".to_string(),
        "  C: 60 to 74".to_string(),
        "  D: Below 60".to_string(),
        "".to_string(),
        "Note: This report card is auto-generated by a Rust-based console application.".to_st
    ];
```

```
    let font_size = 12.0;
    let mut current_y = Mm(280.0);
    for line in lines {
        layer.use_text(line, font_size, Mm(20.0), current_y, &font);
        current_y -= Mm(8.0);
    }
    let file = File::create("full_student_report_card.pdf").unwrap();
    doc.save(&mut BufWriter::new(file)).unwrap();
}

fn main() {
    let mut name = String::new();
    let mut total_str = String::new();
    let mut subjects_str = String::new();

    println!("Enter student name:");
    io::stdin().read_line(&mut name).unwrap();

    println!("Enter total marks:");
    io::stdin().read_line(&mut total_str).unwrap();
    let total: f64 = total_str.trim().parse().unwrap();

    println!("Enter number of subjects:");
    io::stdin().read_line(&mut subjects_str).unwrap();
    let subjects: u32 = subjects_str.trim().parse().unwrap();

    let average = calculate_average(total, subjects);
    let grade = assign_grade(average);

    println!("\n--- Report Card ---");
    println!("Name     : {}", name.trim());
    println!("Total    : {}", total);
    println!("Subjects : {}", subjects);
    println!("Average  : {:.2}", average);
    println!("Grade    : {}", grade);

    generate_pdf(name.trim(), total, subjects, average, grade);
    println!("\nPDF report card generated: full_student_report_card.pdf");
}

Sample Run Output:
Enter student name:
John Doe
Enter total marks:
420
Enter number of subjects:
5

--- Report Card ---
Name     : John Doe
Total    : 420
Subjects : 5
Average  : 84.00
Grade    : B

PDF report card generated: full_student_report_card.pdf
```