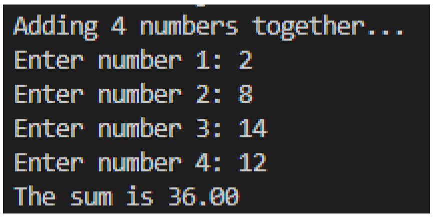
Assignment 4: Functions

Cody Raposa

ELEC2850 Microcontrollers Using C Programming

October 21, 2024

1 Part 1



```
Adding 4 numbers together...
Enter number 1: 2
Enter number 2: 8
Enter number 3: 14
Enter number 4: 12
The sum is 36.00
```

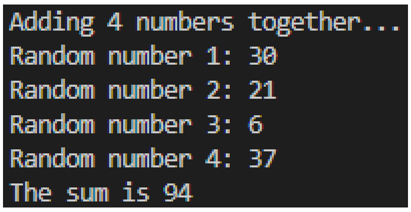
Figure 1: Test cases for Part 1

```c
#include <stdio.h>  // Standard input-output library
#include <stdlib.h> // Standard library
#include <time.h>   // Library for time functions

// Function to sum 4 floating-point numbers entered by the user
float sum4()
{
    printf("Adding 4 numbers together...\n"); // Inform the user about the operation
    float num, sum = 0;                        // Initialize variables for the current number and the sum
    for (int i = 0; i < 4; i++)                // Loop to get 4 numbers from the user
    {
        printf("Enter number %d: ", i + 1); // Prompt the user to enter a number
        scanf("%f", &num);                  // Read the number entered by the user
        sum += num;                         // Add the number to the sum
    }
    printf("The sum is %.2f\n", sum); // Print the sum of the 4 numbers
    return sum;                       // Return the sum
}

int main()
{
    sum4();   // Call the sum4 function
    return 0; // Return 0 to indicate successful execution
}
```

2 Part 3



```
Adding 4 numbers together...
Random number 1: 30
Random number 2: 21
Random number 3: 6
Random number 4: 37
The sum is 94
```

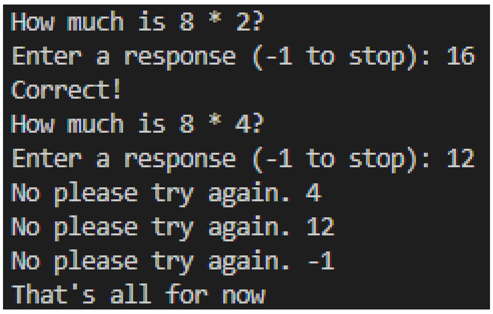
Figure 2: Test cases for Part 3

```c
#include <stdio.h>  // Standard input/output library
#include <stdlib.h> // Standard library for functions like rand()
#include <time.h>   // Library for time functions

// Function to generate and sum 4 random integers
int sum4()
{
    printf("Adding 4 numbers together...\n"); // Inform the user about the operation
    int num, sum = 0; // Initialize variables for the current number and the sum
    for (int i = 0; i < 4; i++) // Loop to generate and sum 4 random numbers
    {
        num = 1 + (int)rand() % 100; // Generate a random number between 1 and 100
        sum += num; // Add the random number to the sum
        printf("Random number %d: %d\n", i+1, num); // Print the generated random number
    }
    return sum; // Return the sum of the 4 random numbers
}

int main()
{
    srand(time(NULL)); // Seed the random number generator with the current time
    int ans = sum4(); // Call the sum4 function and store the result in ans
    printf("The sum is %d\n", ans); // Print the sum of the 4 random numbers
    return 0; // Return 0 to indicate successful execution
}
```

3 Part 4



```
How much is 8 * 2?
Enter a response (-1 to stop): 16
Correct!
How much is 8 * 4?
Enter a response (-1 to stop): 12
No please try again. 4
No please try again. 12
No please try again. -1
That's all for now
```

Figure 3: Test cases for Part 4

```c
#include <stdio.h>  // Standard input-output library
#include <stdlib.h> // Standard library for functions like rand()
#include <time.h>   // Library for time functions

// Function prototype for generating a multiplication problem
int generateProblem(); // returns answer

// Function to generate a multiplication problem and return the answer
int generateProblem()
{
    int num1, num2 = 0; // Initialize two integers
    num1 = 1 + (int)rand() % 10; // Generate a random number between 1 and 10
    num2 = 1 + (int)rand() % 10; // Generate another random number between 1 and 10
    printf("How much is %d * %d? \n", num1, num2); // Print the multiplication problem
    return num1 * num2; // Return the product of the two numbers
}

int main()
{
    srand(time(NULL)); // Seed the random number generator with the current time
    while (1) // Infinite loop to keep generating problems
    {
        int ans = generateProblem(); // Generate a multiplication problem and get the correct answer
        int userAns = 0; // Initialize the user's answer
        printf("Enter a response (-1 to stop): "); // Prompt the user for an answer
        scanf("%d", &userAns); // Read the user's answer
        if (userAns == ans) // Check if the user's answer is correct
        {
            printf("Correct!\n"); // Inform the user that the answer is correct
        }
        while (userAns != ans) // Loop until the user provides the correct answer
        {
            if (userAns == -1) // Check if the user wants to stop
            {
                break; // Exit the inner loop
            }
            printf("No please try again. "); // Inform the user that the answer is incorrect
            scanf("%d", &userAns); // Read the user's new answer
            if (userAns == ans) // Check if the new answer is correct
            {
                printf("Correct!\n"); // Inform the user that the answer is correct
            }
        }
        if (userAns == -1) // Check if the user wants to stop
        {
            break; // Exit the outer loop
        }
    }
    printf("That's all for now\n"); // Inform the user that the program is ending
    return 0; // Return 0 to indicate successful execution
}
```

4 Part 5 Problem Statement

Using the code from part 4, modify it to vary the resssponses for a correct and incorrect response, using functions.

5 Analysis

5.1 Inputs

Answer (int)

6 Flowchart

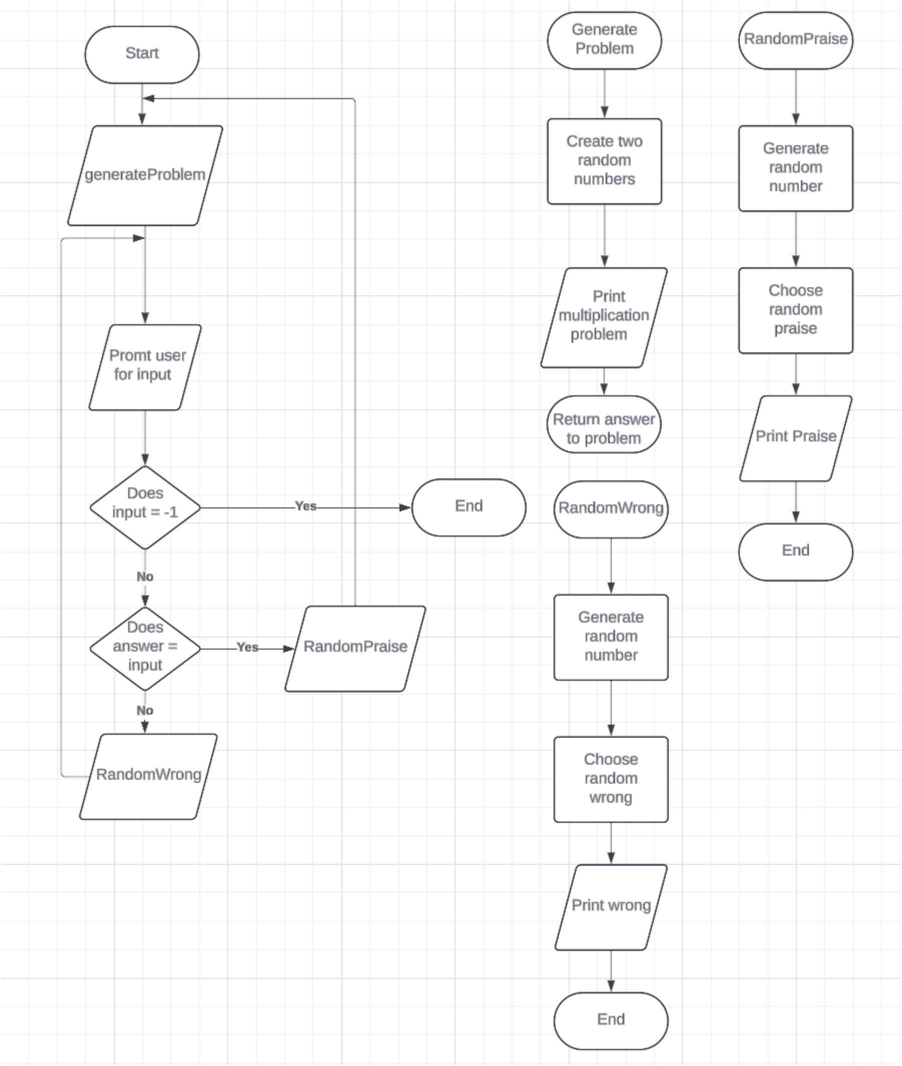
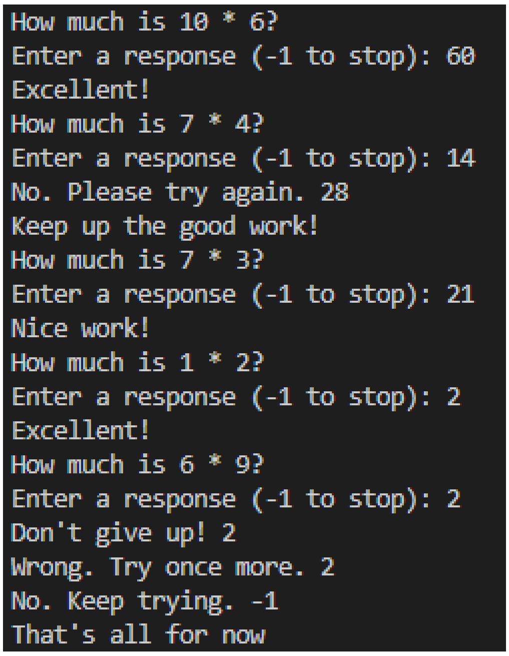


Figure 4: Flowchart for Q2

7 Output



```
How much is 10 * 6?
Enter a response (-1 to stop): 60
Excellent!
How much is 7 * 4?
Enter a response (-1 to stop): 14
No. Please try again. 28
Keep up the good work!
How much is 7 * 3?
Enter a response (-1 to stop): 21
Nice work!
How much is 1 * 2?
Enter a response (-1 to stop): 2
Excellent!
How much is 6 * 9?
Enter a response (-1 to stop): 2
Don't give up! 2
Wrong. Try once more. 2
No. Keep trying. -1
That's all for now
```

Figure 5: A test case for the program

8 Code

```
#include <stdio.h>  // Standard input/output library
#include <stdlib.h> // Standard library for functions like rand()
#include <time.h>   // Library for the time function

// Function prototypes
int generateProblem(); // returns answer
void randomPraise();   // prints a random praise message
void randomWrong();    // prints a random wrong message

// Function to generate a multiplication problem
int generateProblem()
{
    int num1, num2 = 0;                              // Initialize num1 and num2 to 0
    num1 = 1 + (int)rand() % 10;                     // Generate a random number between 1 and 10
    num2 = 1 + (int)rand() % 10;                     // Generate another random number between 1 and
    10
    printf(”How much is %d ∗ %d? \n”, num1, num2); // Print the multiplication problem
    return num1 ∗ num2;                              // Return the answer to the problem
}

// Function to print a random praise message
void randomPraise()
{
    int random = rand() % 4; // Generate a random number between 0 and 3
    switch (random)
    {
    case 0:
        printf(”Very Good!\n”); // Print ”Very Good!” if random is 1
        break;
    case 1:
        printf(”Excellent!\n”); // Print ”Excellent!” if random is 2
        break;
    case 2:
        printf(”Nice work!\n”); // Print ”Nice work!” if random is 3
        break;
    case 3:
        printf(”Keep up the good work!\n”); // Print ”Keep up the good work!” if random is 4
        break;
    default:
        printf(”Very good!\n”); // Default case to print ”Very good!”, shouldn’t be reached
        break;
    }
}
```

```c

// Function to print a random wrong message
void randomWrong()
{
    int random = rand() % 4;
    switch (random)
    {
    case 0:
        printf("No. Please try again. "); // Print "No. Please try again." if random is 0
        break;
    case 1:
        printf("Wrong. Try once more. "); // Print "Wrong. Try once more." if random is 1
        break;
    case 2:
        printf("Don't give up! "); // Print "Don't give up!" if random is 2
        break;
    case 3:
        printf("No. Keep trying. "); // Print "No. Keep trying." if random is 3
        break;
    default:
        printf("No. Please try again. "); // Default case to print "No. Please try again.", shouldn't be reached
        break;
    }
}

int main()
{
    srand(time(NULL)); // Seed the random number generator with the current system time
    while (1)          // Infinite loop to keep generating problems until user enters -1
    {
        int ans = generateProblem();               // Generate a multiplication problem and store the answer
        int userAns = 0;                           // Initialize userAns to 0
        printf("Enter a response (-1 to stop): "); // Prompt the user to enter an answer
        scanf("%d", &userAns);                     // Read the user's answer
        if (userAns == ans)                        // Check if the user's answer is correct
        {
            randomPraise(); // Print a random praise message
        }
        while (userAns != ans) // Loop until the user enters the correct answer or -1 to stop
        {
            if (userAns == -1)
            {
                break; // Break out of the loop if the user enters -1
            }
            randomWrong();         // Print a random wrong message
            scanf("%d", &userAns); // Read the user's answer again
            if (userAns == ans)    // Check if the user's answer is correct
            {
                randomPraise(); // Print a random praise message
            }
        }
        if (userAns == -1) // Check if the user entered -1 to stop
        {
            break; // Break out of the infinite loop
        }
    }
    printf("That's all for now\n"); // Print a message to indicate the end of the program
    return 0;                       // Return 0 to indicate successful completion of the program
}
```