Lab Assignment 3
Loops

COL100

22 March 2021

1 Palindrome Checker

Write a C program to input a number and find if the number is a palindrome. Palindromic numbers read the same from left to right and right to left. If the number is a palindrome, then output "Palindrome"; otherwise, output "Not Palindrome". You may assume the input to be a positive integer.

Example

1)
Input

```
9
```

Output

```
Palindrome
```

2)
Input

```
19291
```

Output

```
Palindrome
```

3)
Input

```
2945
```

Output

```
Not Palindrome
```

2 Arithmetic Progression

Write a C program to print the first n terms of an arithmetic progression given the first term a and the common difference d. The inputs to the program are $n, a,$ and d, respectively. Assume n ($n > 1$)is a positive integer and a and d could be real-valued.

Example

1)
Input

```
6 7 2
```

Output

```
7 9 11 13 15 17
```

2)
Input

```
5 11 -3.5
```

Output

```
11 7.5 4 0.5 -3
```

3 Sum of Numbers

Write a C program to do the following task. Take a positive integer n as input. Then take n numbers as input and print their sum. These numbers could be real-valued.

Example

Input

```
4
1.3 3 5.3 -2.1
```

Output

```
7.5
```

4 Finding Cosine

In assignment 2, you computed the factorial ($x!$) of a number and the power of a number (x^n). Using that knowledge, can you write a C program to compute the value of $cos(x)$ for a given input x? Note that, x here is given in degrees. You need to convert it to radians before using the appropriate formula for computing the cosine. Assume $\pi = 3.14$, x to be real-valued, and $0 \leq x \leq 180$. Print the value of $cos(x)$ upto 3 decimal places.
(*Hint*: Think about the Taylor series expansion of $cos(x)$. Consider upto the term $\frac{x^{12}}{12!}$.)

Example

1)
Input

```
45
```

Output

```
0.707
```

2)
Input

```
30
```

Output

```
0.866
```

3)
Input

```
150
```

Output

```
-0.866
```

5 Special Sum

Write a C program to find the sum of all numbers divisible by 17 within 100 to 999. It is easy with the modulus(%) operator. Can you do it without using the modulus operator?

6 Armstrong Numbers

Write a C program to take a positive integer n as input. Print all the Armstrong numbers from 1 to n. You may assume $n \leq 10000$.
A k-digit number is Armstrong if the sum of the digits raised to k is equal to the number itself. For example, a 3-digit number will be Armstrong if the sum of the cubes of the digits is equal to the number itself (153 is an Armstrong number as $153 = 1^3 + 5^3 + 3^3$).

Example

1)
Input

```
200
```

Output

```
1 2 3 4 5 6 7 8 9 153
```

2)
Input

```
400
```

Output

```
1 2 3 4 5 6 7 8 9 153 370 371
```

7 GCD of Two Numbers

Write a C program to find the GCD of two given numbers. Assume both the numbers to be distinct integers and greater than 1.

Example

1)
Input

```
81 153
```

Output

```
9
```

2)
Input

```
35 21
```

Output

```
7
```

8 Perfect Number Checker

Write a C program to take a number as input and check if it is a perfect number or not. A perfect number is equal to the sum of all its factors except itself. For example, 6 is a perfect number as $6 = 1 + 2 + 3$.
Assume the input to be a non-zero positive integer. Your program should output "Perfect" if the number is perfect; otherwise, output "Not Perfect".

Example

1)
Input

```
28
```

Output

```
Perfect
```

$(28 = 1 + 2 + 4 + 7 + 14)$

2)
Input

```
21
```

Output

```
Not Perfect
```

$(21 \neq 1 + 3 + 7)$

Submission and other logistics

Submit at least 4 code solutions(.c files of 4 questions) as a **zip** file on Gradescope (to your respective group's course). Additionally, add screenshots in the same submission showing the execution of your code on your terminal with outputs for some given inputs. Submit only one .c file for each question. Use separate .c files for each new question. Please name your .c files as per the question number (**q1.c, q2.c, ...** etc). Following this naming convention will help TAs to figure out where to look the answers easily. You can also submit more than 4 or all questions to increase your chances of full marks.

Example: To zip folder 'a3' as 'a3.zip':

```
zip -r a3.zip a3
```

It is highly **recommended** that you name the variables with proper names as per the question to identify them quickly. Comments in your codes are also highly **encouraged** and make life easier for everyone.
You can check the **2nd Chapter** in NASAs C style guide for styling recommendations.

You can work either individually or with another student of your group for this assignment. **Only one** submission on Gradescope is enough for a team, but you need to **add your teammate** on Gradescope after submission. Follow these steps for adding your team member.
Note: you can change your team for future assignments