Some Tough Checking

1. **WAP in C to display prime numbers from 1 to 20.**

```
// A prime number has exactly two positive divisors i.e 1 and itself

#include <stdio.h>

int main()
{
        int num, i, count;

        for(num = 1; num <= 20; num++)
        {
                count = 0;

                for(i = 1; i <= num; i++)
                {
                        if(num % i == 0){
                                count ++;
                        }
                }

                if (count == 2) {
                        printf("\n %d", num);
                }
        }

        getch();
        return 0;
}
```

2. WAP in C to find the sum of individual digit given by the user.

```
#include <stdio.h>
#include <math.h>

int main()
{

    int num, digit, sum = 0;

    printf("Enter a number: ");
    scanf("%d", &num);

           //    num = abs(num);

    while (num != 0)
           {
        digit = num % 10;           // extracting only the last digit from num
        sum = sum + digit;          // Add the digit to the sum
        num = num / 10;             // removing only the last digit from num
    }

    printf("\nSum of digits = %d \n", sum);

    getch();
    return 0;
}
```

3. **WAP in C to check whether the given number is palindrome or not.**

```
#include <stdio.h>

int main() {

        int num, originalNum, remainder, reverse = 0;

        printf("Enter an integer: ");
        scanf("%d", &num);

        originalNum = num;

        while (num != 0) {
                remainder = num % 10;                    // extracting only the last digit from num
                reverse = reverse * 10 + remainder;
                num = num / 10;                          // removing the last digit from num
        }

        if (originalNum == reverse)
        {
                printf("\n %d is a palindrome number.\n", originalNum);}
        else {
                printf("\n %d is not a palindrome number.\n", originalNum);
        }

        getch();
        return 0;
}
```

4. **Write a C program to enter a 3 digit number and check whether the given number is armstrong or not.**

```
#include <stdio.h>
#include <math.h>

int main()
{
        int num, originalNum, remainder;
        int result = 0;

        printf("Enter a 3-digit number: ");
        scanf("%d", &num);

        originalNum = num;

        if (num < 100 || num > 999) {
                printf("Only allowed to enter 3-digit number !!\n");
                return 1;
        }

        while (num != 0)
        {
                remainder = num % 10;                   // extracting only the last digit from num
                result = result +  pow(remainder, 3);   // Adding cube of remainder to result
                num = num / 10;                         // extracting only the last digit from num
        }


        if (originalNum == result) {
                printf("\n %d is an Armstrong number.\n", originalNum);
        }
        else {
                printf("\n %d is not an Armstrong number.\n", originalNum);
        }

        getch();
        return 0;
}
```

5. **To check null terminated character (i.e \0) in the string.**

```
#include <stdio.h>
#include <string.h>

int main()
{
        int i;
        char str[] = "NEPAL";

        for(i=0; i <= strlen(str); i++)
        {
                printf("\n str[%d]: %c \n", i, str[i]);

                if (str[i] == '\0'){
                        printf("\n Null character is found at str[%d]", i);
                }
        }

        getch();
        return 0;
}
```

6. Asdf