Supplementary programs for understanding character and string handling in C Language

This document contains three programs that will help you to understand the example code in the document C Characters and Strings.

1. Following program intends to show how char variables can be manipulated:

```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    char grade = 'A';

    printf("\n************   OUTPUT **********************");
    printf("\n\n");

    printf("\n Character stored in grade variable is \t %c", grade);

    printf("\n int value represented by char variable grade is \t %d", grade);

    grade = 66;

    printf("\n new Character value of  grade variable is \t %c", grade);

    grade++;


    printf("\n Character value of  grade variable after increment \t %c", grade);

    printf("\n\n");

    printf("************   ****** **********************\n");

    return 0;
}
```

```
************   OUTPUT **********************

 Character stored in grade variable is   A
 int value represented by char variable grade is     65
 new Character value of  grade variable is   B
 Character value of  grade variable after increment      C

************   ****** **********************
```

2. Character Arrays, and pointer to char

```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{
    char course[] = "Programming Fundamentals";

    char course2[] = {'D', 'B', 'M', 'S', '\0'};

    const char *instructor = "Imran Ali";

    printf("\n************   OUTPUT **********************");

    printf("\n\n");

    printf("\n value stored in course character Array %s ", course);

    printf("\n value stored in course2 character Array %s ", course2);

    printf("\n value stored in pointer variable %s ", instructor);

    printf("\n\n");

    printf("************   ****** **********************\n");

    return 0;
}
```

```
************   OUTPUT **********************

 value stored in course character Array Programming Fundamentals
 value stored in course2 character Array DBMS
 value stored in pointer variable Imran Ali

************   ****** **********************
```

3. Use of Ternary Operator

```c
#include <stdio.h>
#include <ctype.h>

int main(void)
{


    printf("\n************   OUTPUT *********************");

    printf("\n\n");

    printf((2 > 3) ? "true" : "false");

    printf("\n");

    printf((2 < 3) ? "true" : "false");

    printf("\n\n");

    printf("************   ****** **********************\n");

    return 0;
}
```

```
************   OUTPUT **********************

false
true

************   ****** **********************
```