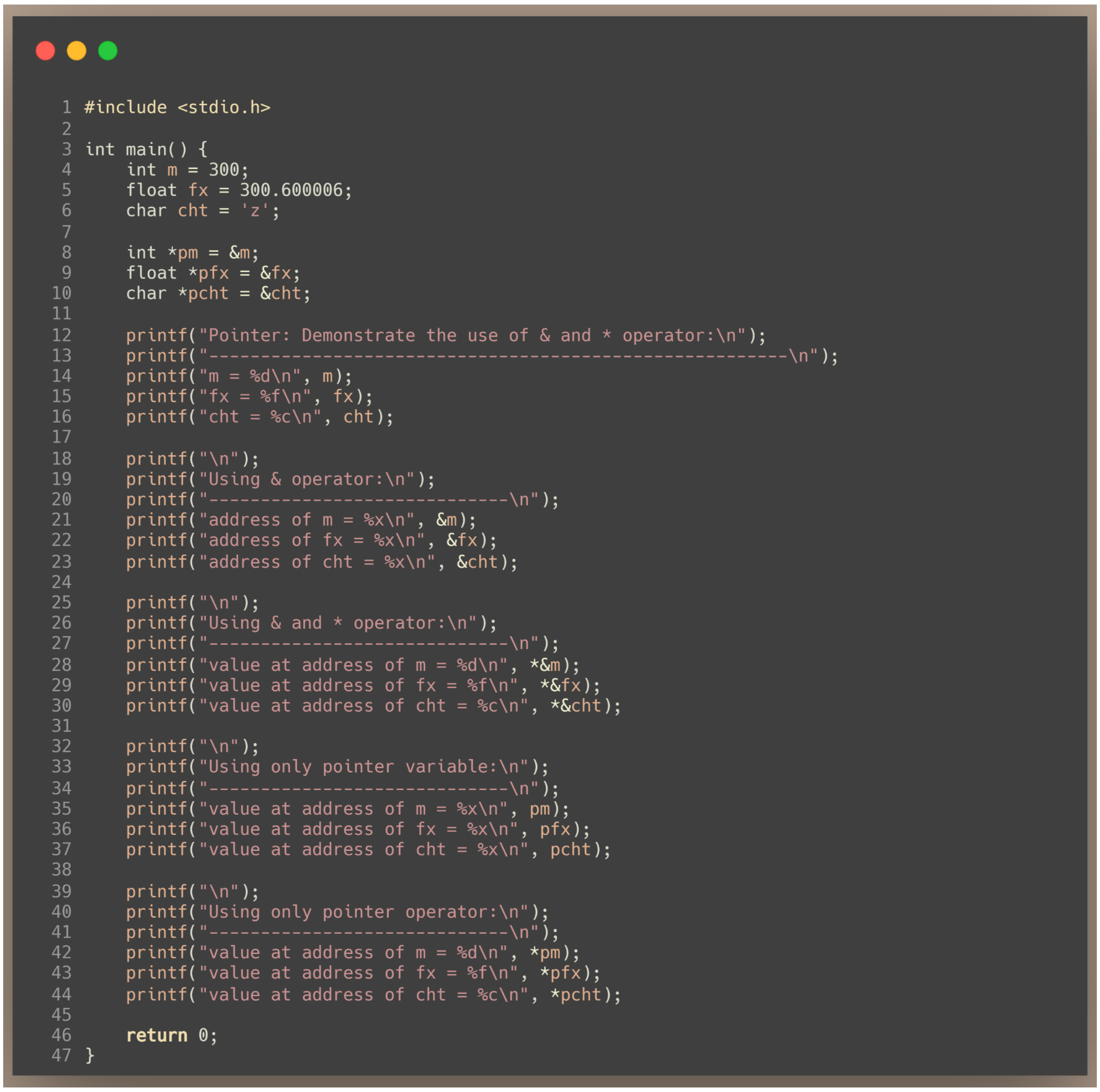
Name: Mark Achiles G. Flores Jr. **Function without Arguments**
Year & Section: BSCS 1-4

1. Write a program in C to demonstrate the use of & (address of) and *(value at address) operator.
   Expected Output:

```
Pointer: Demonstrate the use of & and * operator:
--------------------------------------------------------
m = 300
fx = 300.600006
cht = z

Using & operator:
-----------------------
address of m = 0x7ffda2eeeec8
address of fx = 0x7ffda2eeeecc
address of cht = 0x7ffda2eeeec7

Using & and * operator:
-----------------------------
value at address of m = 300
value at address of fx = 300.600006
value at address of cht = z

Using only pointer variable:
----------------------------------
address of m = 0x7ffda2eeeec8
address of fx = 0x7ffda2eeeecc
address of cht = 0x7ffda2eeeec7

Using only pointer operator:
----------------------------------
value at address of m = 300
value at address of fx= 300.600006
value at address of cht= z
```

Name: Mark Achiles G. Flores Jr. **Function without Arguments**

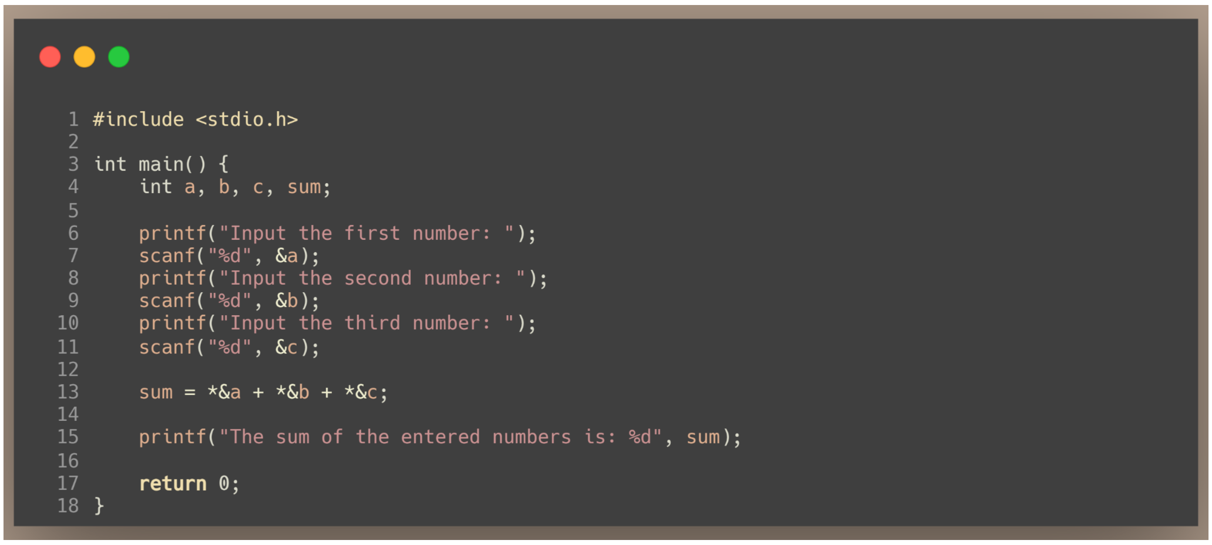
Year & Section: BSCS 1-4



```c
#include <stdio.h>

int main() {
    int m = 300;
    float fx = 300.600006;
    char cht = 'z';

    int *pm = &m;
    float *pfx = &fx;
    char *pcht = &cht;

    printf("Pointer: Demonstrate the use of & and * operator:\n");
    printf("--------------------------------------------------------\n");
    printf("m = %d\n", m);
    printf("fx = %f\n", fx);
    printf("cht = %c\n", cht);

    printf("\n");
    printf("Using & operator:\n");
    printf("-----------------------------\n");
    printf("address of m = %x\n", &m);
    printf("address of fx = %x\n", &fx);
    printf("address of cht = %x\n", &cht);

    printf("\n");
    printf("Using & and * operator:\n");
    printf("-----------------------------\n");
    printf("value at address of m = %d\n", *&m);
    printf("value at address of fx = %f\n", *&fx);
    printf("value at address of cht = %c\n", *&cht);

    printf("\n");
    printf("Using only pointer variable:\n");
    printf("-----------------------------\n");
    printf("value at address of m = %x\n", pm);
    printf("value at address of fx = %x\n", pfx);
    printf("value at address of cht = %x\n", pcht);

    printf("\n");
    printf("Using only pointer operator:\n");
    printf("-----------------------------\n");
    printf("value at address of m = %d\n", *pm);
    printf("value at address of fx = %f\n", *pfx);
    printf("value at address of cht = %c\n", *pcht);

    return 0;
}
```

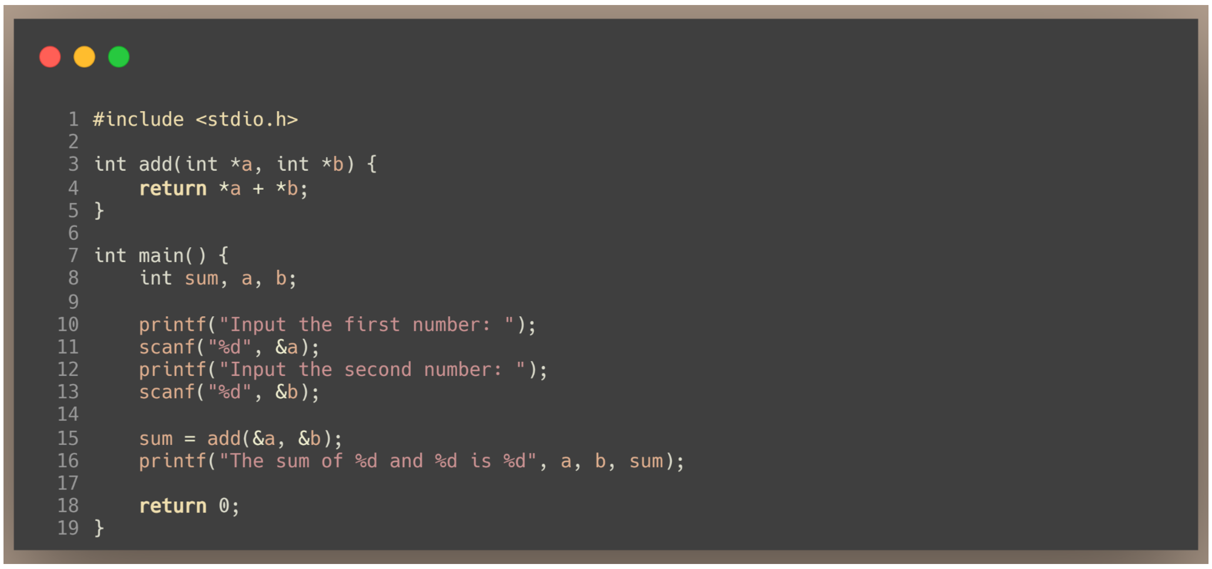
Name: Mark Achiles G. Flores Jr. **Function without Arguments**
Year & Section: BSCS 1-4

2. Write a program in C to add three numbers using pointers.
   Test Data:
   Input the first number: 5
   Input the second number: 10
   Input the third number: 15
   Expected Output:
   The sum of the entered numbers is: 30



```c
#include <stdio.h>

int main() {
    int a, b, c, sum;

    printf("Input the first number: ");
    scanf("%d", &a);
    printf("Input the second number: ");
    scanf("%d", &b);
    printf("Input the third number: ");
    scanf("%d", &c);

    sum = *&a + *&b + *&c;

    printf("The sum of the entered numbers is: %d", sum);

    return 0;
}
```

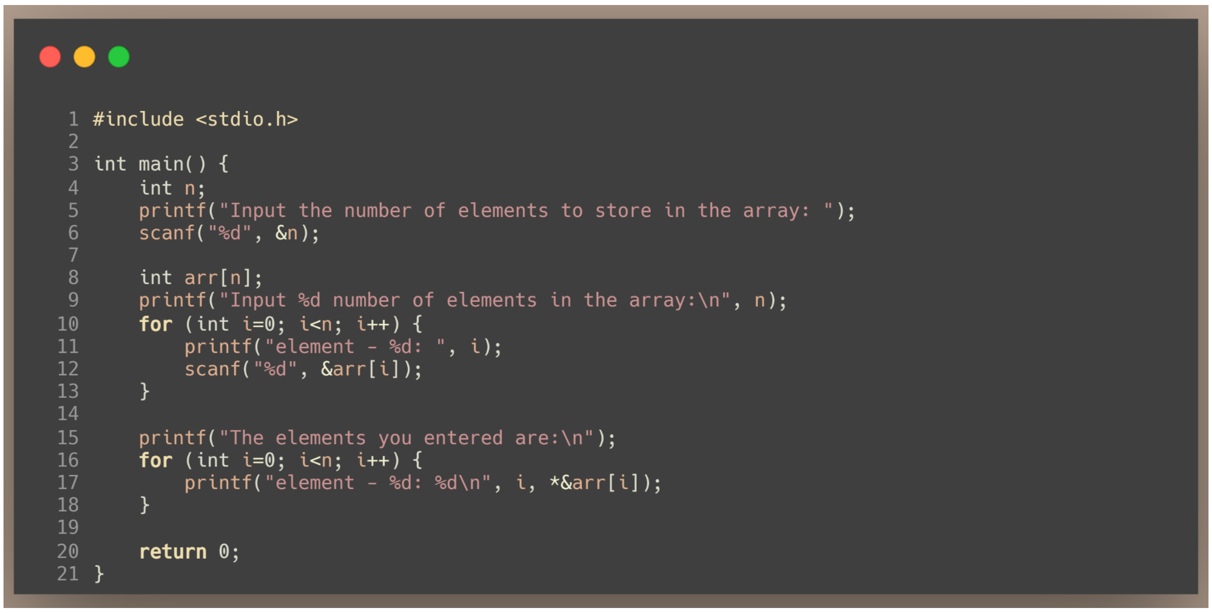
3. Write a program in C to add numbers using call by reference.
   Test Data:
   Input the first number: 5
   Input the second number: 6
   Expected Output:
   The sum of 5 and 6 is 11



```c
#include <stdio.h>

int add(int *a, int *b) {
    return *a + *b;
}

int main() {
    int sum, a, b;

    printf("Input the first number: ");
    scanf("%d", &a);
    printf("Input the second number: ");
    scanf("%d", &b);

    sum = add(&a, &b);
    printf("The sum of %d and %d is %d", a, b, sum);

    return 0;
}
```

Name: Mark Achiles G. Flores Jr. **Function without Arguments**
Year & Section: BSCS 1-4

4. Write a program in C to store n elements in an array and print the elements using pointer.
   Test Data:
   Input the number of elements to store in the array :5
   Input 5 number of elements in the array:
   element - 0: 5
   element - 1: 7
   element - 2: 2
   element - 3: 9
   element - 4: 8
   Expected Output:
   The elements you entered are:
   element - 0: 5
   element - 1: 7
   element - 2: 2
   element - 3: 9
   element - 4: 8

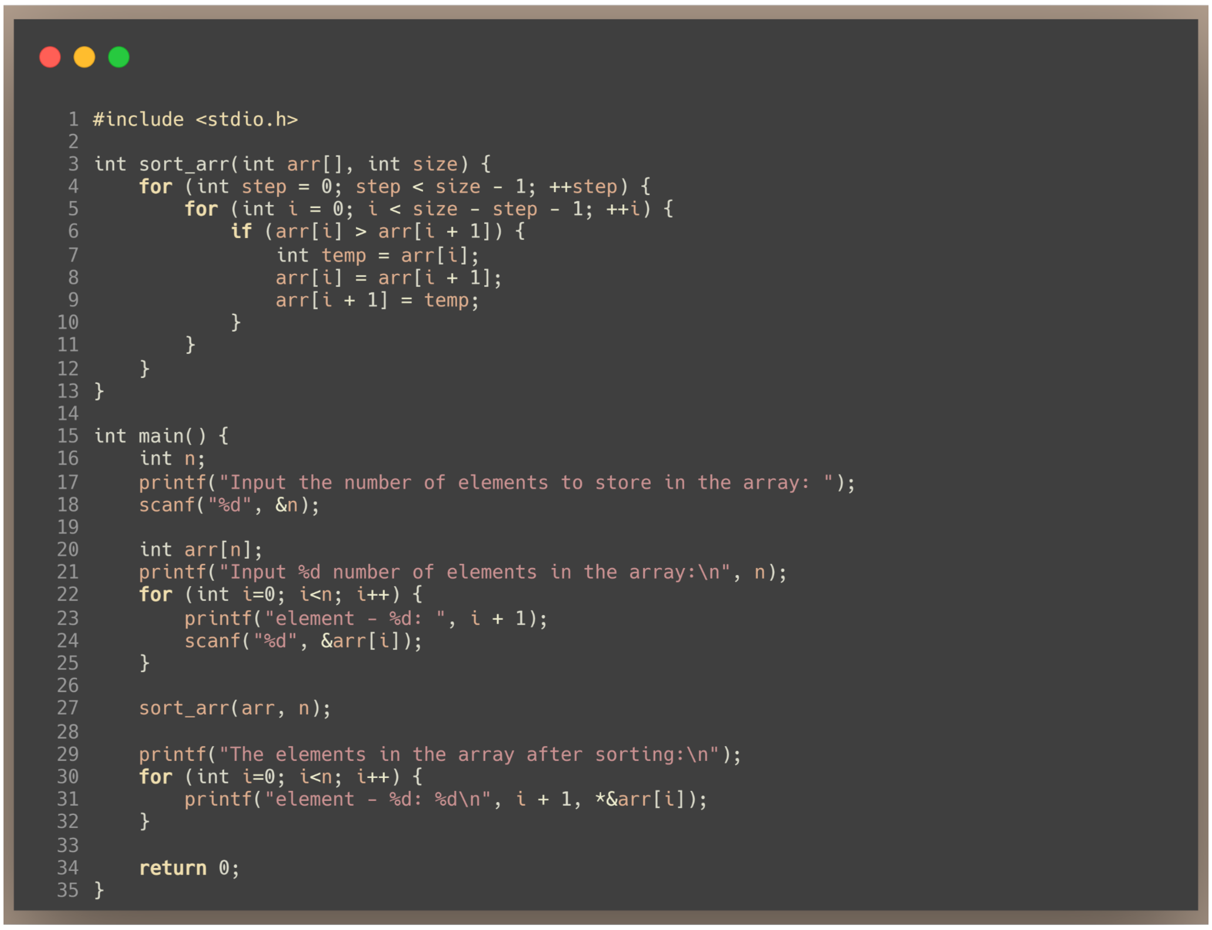


```c
#include <stdio.h>

int main() {
    int n;
    printf("Input the number of elements to store in the array: ");
    scanf("%d", &n);

    int arr[n];
    printf("Input %d number of elements in the array:\n", n);
    for (int i=0; i<n; i++) {
        printf("element - %d: ", i);
        scanf("%d", &arr[i]);
    }

    printf("The elements you entered are:\n");
    for (int i=0; i<n; i++) {
        printf("element - %d: %d\n", i, *&arr[i]);
    }

    return 0;
}
```

Name: Mark Achiles G. Flores Jr. **Function without Arguments**
Year & Section: BSCS 1-4

5. Write a program in C to sort an array using Pointer.
   Test Data:
   Input the number of elements to store in the array: 5
   Input 5 number of elements in the array:
   element - 1: 25
   element - 2: 45
   element - 3: 89
   element - 4: 15
   element - 5: 82

   Expected Output:
   The elements in the array after sorting:
   element - 1: 15
   element - 2: 25
   element - 3: 45
   element - 4: 82
   element - 5: 89



```c
#include <stdio.h>

int sort_arr(int arr[], int size) {
    for (int step = 0; step < size - 1; ++step) {
        for (int i = 0; i < size - step - 1; ++i) {
            if (arr[i] > arr[i + 1]) {
                int temp = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = temp;
            }
        }
    }
}

int main() {
    int n;
    printf("Input the number of elements to store in the array: ");
    scanf("%d", &n);

    int arr[n];
    printf("Input %d number of elements in the array:\n", n);
    for (int i=0; i<n; i++) {
        printf("element - %d: ", i + 1);
        scanf("%d", &arr[i]);
    }

    sort_arr(arr, n);

    printf("The elements in the array after sorting:\n");
    for (int i=0; i<n; i++) {
        printf("element - %d: %d\n", i + 1, *&arr[i]);
    }

    return 0;
}
```