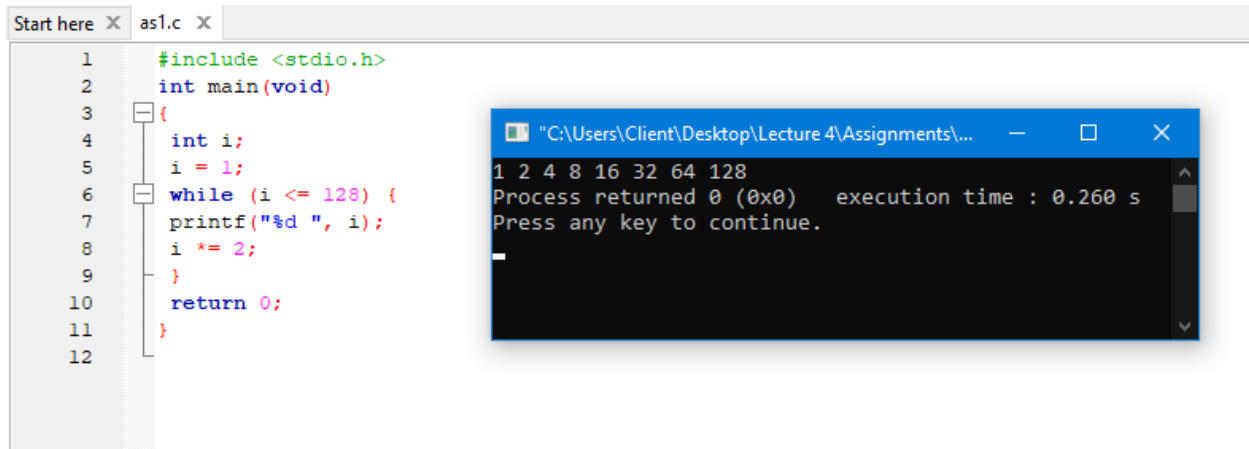


CMSC 21 – Lecture 4 Assignment

Francis Wilfred B. Olilang

1.

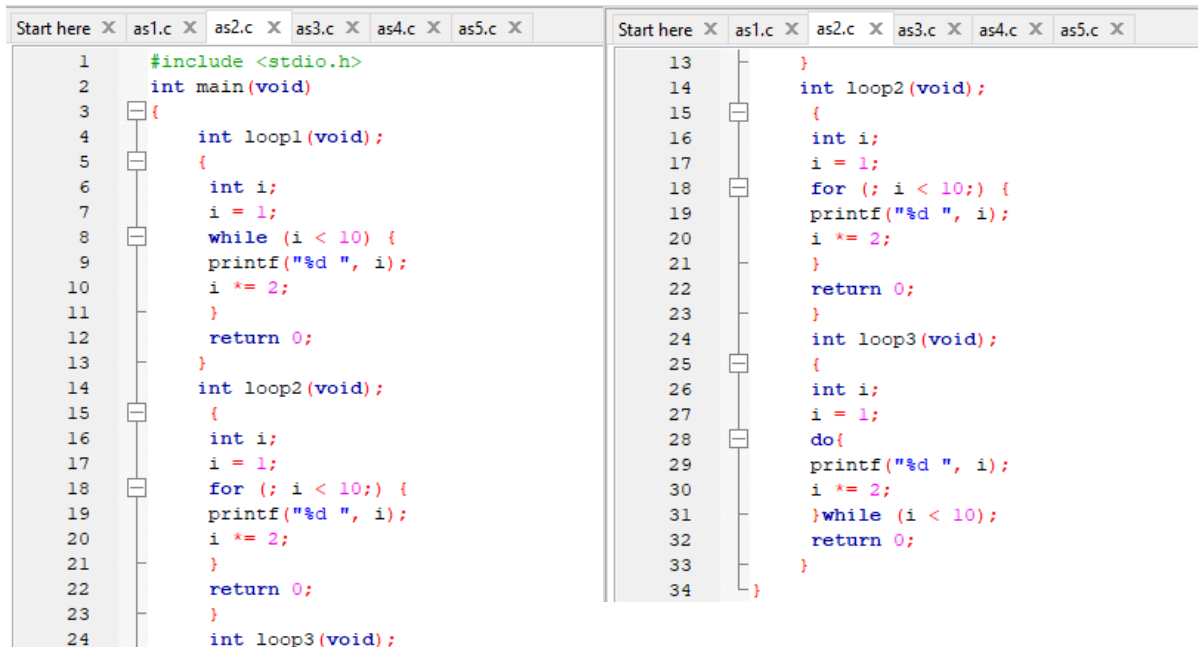


The screenshot shows a C program in a text editor and its execution output in a terminal window. The program, named `as1.c`, includes `<stdio.h>` and defines a `main` function. Inside `main`, it declares an integer `i`, initializes it to 1, and enters a `while` loop that continues as long as `i` is less than or equal to 128. In each iteration, it prints the value of `i` followed by a space and then doubles `i` by multiplying it by 2. After the loop, it returns 0. The terminal window shows the output: `1 2 4 8 16 32 64 128`, followed by the message "Process returned 0 (0x0) execution time : 0.260 s" and a prompt to press any key to continue.

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int i;
5     i = 1;
6     while (i <= 128) {
7         printf("%d ", i);
8         i *= 2;
9     }
10    return 0;
11 }
12
```

```
1 2 4 8 16 32 64 128
Process returned 0 (0x0) execution time : 0.260 s
Press any key to continue.
```

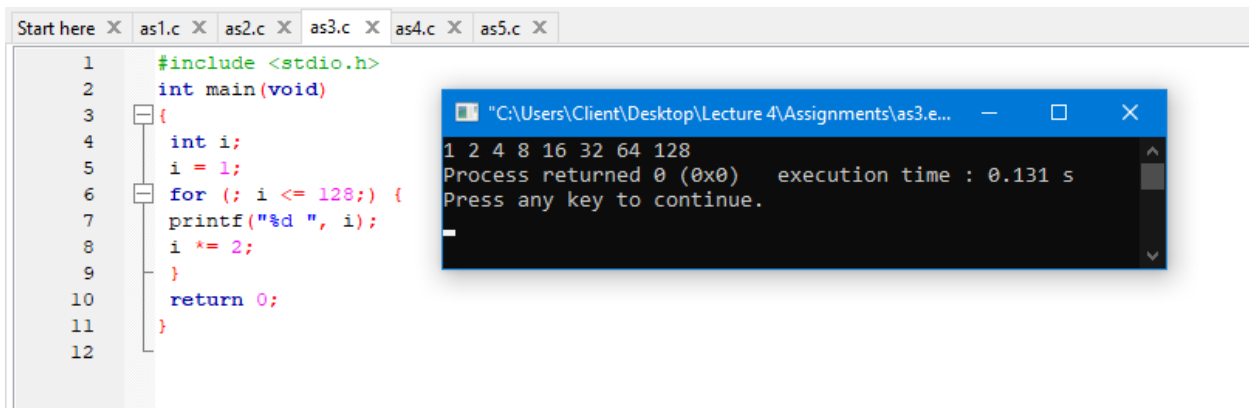
2. All statements are equivalent.



The screenshot displays two C programs side-by-side in a text editor. The left program, `as1.c`, is identical to the one in the first screenshot. The right program, `as2.c`, defines two functions: `loop2` and `loop3`. `loop2` is a `void` function that declares an integer `i`, initializes it to 1, and enters a `for` loop that runs while `i` is less than 10, printing `i` and doubling it in each iteration. `loop3` is another `void` function that does the same thing but uses a `do-while` loop instead of a `for` loop. Both functions return 0. The main function in `as2.c` calls `loop2` and `loop3`.

```
1 #include <stdio.h>
2 int main(void)
3 {
4     int loop1(void);
5     {
6         int i;
7         i = 1;
8         while (i < 10) {
9             printf("%d ", i);
10            i *= 2;
11        }
12        return 0;
13    }
14    int loop2(void);
15    {
16        int i;
17        i = 1;
18        for (; i < 10;) {
19            printf("%d ", i);
20            i *= 2;
21        }
22        return 0;
23    }
24    int loop3(void);
25    {
26        int i;
27        i = 1;
28        do{
29            printf("%d ", i);
30            i *= 2;
31        }while (i < 10);
32        return 0;
33    }
34 }
```

3.



The screenshot shows a C program in a text editor and its execution output in a terminal window. The program, named `as3.c`, includes `<stdio.h>` and defines a `main` function. Inside `main`, it declares an integer `i`, initializes it to 1, and enters a `for` loop that continues as long as `i` is less than or equal to 128. In each iteration, it prints the value of `i` followed by a space and then doubles `i` by multiplying it by 2. After the loop, it returns 0. The terminal window shows the output: `1 2 4 8 16 32 64 128`, followed by the message "Process returned 0 (0x0) execution time : 0.131 s" and a prompt to press any key to continue.

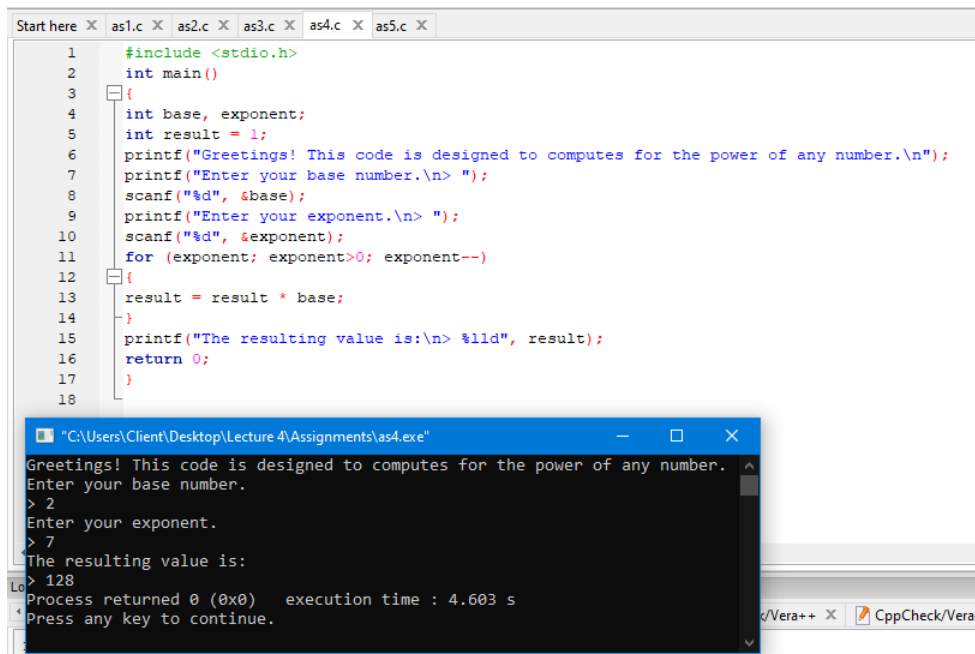
```
1 #include <stdio.h>
2 int main(void)
3 {
4     int i;
5     i = 1;
6     for (; i <= 128;) {
7         printf("%d ", i);
8         i *= 2;
9     }
10    return 0;
11 }
12
```

```
1 2 4 8 16 32 64 128
Process returned 0 (0x0) execution time : 0.131 s
Press any key to continue.
```

CMSC 21 – Lecture 4 Assignment

Francis Wilfred B. Olilang

4.

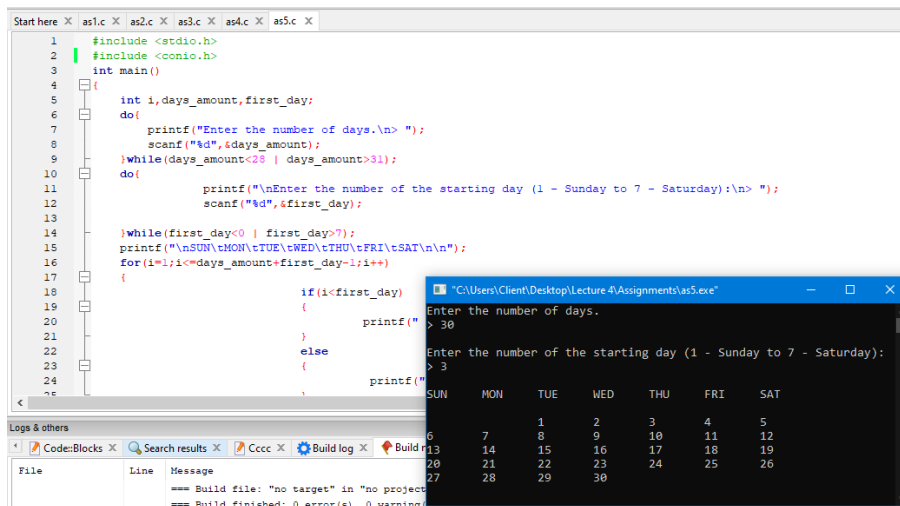


```
1 #include <stdio.h>
2 int main()
3 {
4     int base, exponent;
5     int result = 1;
6     printf("Greetings! This code is designed to computes for the power of any number.\n");
7     printf("Enter your base number.\n> ");
8     scanf("%d", &base);
9     printf("Enter your exponent.\n> ");
10    scanf("%d", &exponent);
11    for (exponent; exponent>0; exponent--)
12    {
13        result = result * base;
14    }
15    printf("The resulting value is:\n> %lld", result);
16    return 0;
17 }
18
```

Execution Output:

```
Greetings! This code is designed to computes for the power of any number.
Enter your base number.
> 2
Enter your exponent.
> 7
The resulting value is:
> 128
Process returned 0 (0x0)   execution time : 4.603 s
Press any key to continue.
```

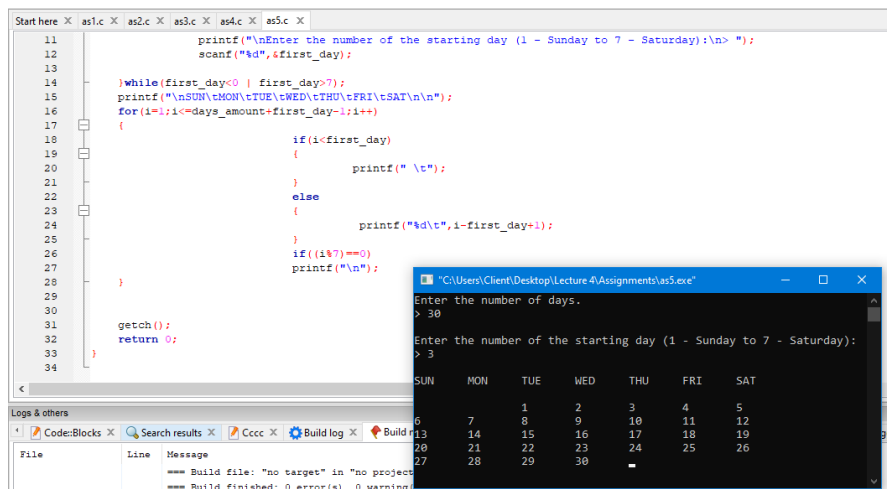
5.



```
1 #include <stdio.h>
2 #include <conio.h>
3 int main()
4 {
5     int i, days_amount, first_day;
6     do{
7         printf("Enter the number of days.\n> ");
8         scanf("%d", &days_amount);
9     }while(days_amount<28 | days_amount>31);
10    do{
11        printf("\nEnter the number of the starting day (1 - Sunday to 7 - Saturday):\n> ");
12        scanf("%d", &first_day);
13    }while(first_day<0 | first_day>7);
14    printf("\nSUN\tMON\tTUE\tWED\tTHU\tFRI\tSAT\n\n");
15    for(i=1; i<=days_amount+first_day-1; i++)
16    {
17        if(i<first_day)
18        {
19            printf(" ");
20        }
21        else
22        {
23            printf("%d\t", i-first_day+1);
24        }
25    }
26    printf("\n");
27 }
28
```

Execution Output:

```
Enter the number of days.
> 30
Enter the number of the starting day (1 - Sunday to 7 - Saturday):
> 3
SUN  MON  TUE  WED  THU  FRI  SAT
6    7    8    9    10   11   12
13   14   15   16   17   18   19
20   21   22   23   24   25   26
27   28   29   30
```



```
11    printf("\nEnter the number of the starting day (1 - Sunday to 7 - Saturday):\n> ");
12    scanf("%d", &first_day);
13
14    }while(first_day<0 | first_day>7);
15    printf("\nSUN\tMON\tTUE\tWED\tTHU\tFRI\tSAT\n\n");
16    for(i=1; i<=days_amount+first_day-1; i++)
17    {
18        if(i<first_day)
19        {
20            printf(" ");
21        }
22        else
23        {
24            printf("%d\t", i-first_day+1);
25        }
26        if((i%7)==0)
27            printf("\n");
28    }
29
30    getch();
31    return 0;
32 }
33
```

Execution Output:

```
Enter the number of days.
> 30
Enter the number of the starting day (1 - Sunday to 7 - Saturday):
> 3
SUN  MON  TUE  WED  THU  FRI  SAT
6    7    8    9    10   11   12
13   14   15   16   17   18   19
20   21   22   23   24   25   26
27   28   29   30
```