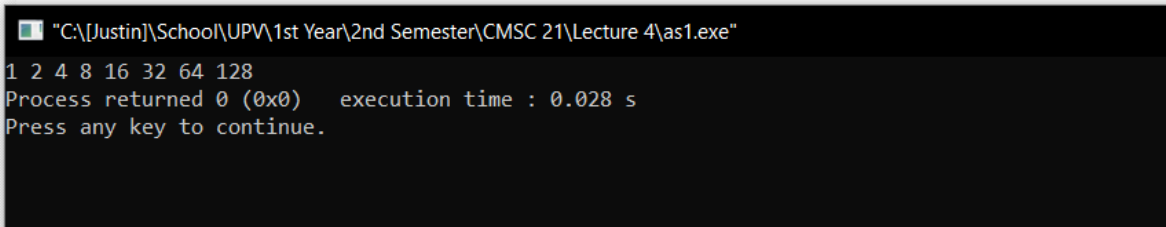


1. The output of the program is "1 2 4 8 16 32 64 128"

Code and Output:

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int i;
5
6      i = 1;
7
8      while (i <= 128) {
9          printf("%d ", i);
10         i *= 2;
11     }
12
13     return 0;
14 }
15
```



2. All three statements are equivalent when i as a starting variable is less than 10. However, when i as a starting variable is greater than or equal to 10, then only statements A and B are equivalent because they will not execute unless i is less than 10. A and B will have no output.

Since C is a do-while statement, it will first execute the loop body, produce an output, and then evaluate i. Therefore, C is not equivalent to A and B.

Example code and output showing equivalence between A, B, C:

### Statement A:

The screenshot shows a C program in a text editor with the following code:

```
1 #include <stdio.h>
2
3 int main(){
4
5     int i;
6
7     i = 1;
8
9     while (i<10){
10
11         printf("%d ", i);
12         i++;
13     }
14
15 }
16
```

Below the code is a terminal window titled "C:\[Justin]\School\UPV\1st Year\2nd Semester\CMSC 21\Lecture 4\as2(Statement A).exe". The terminal output is:

```
1 2 3 4 5 6 7 8 9
Process returned 0 (0x0)   execution time : 0.027 s
Press any key to continue.
```

### Statement B:

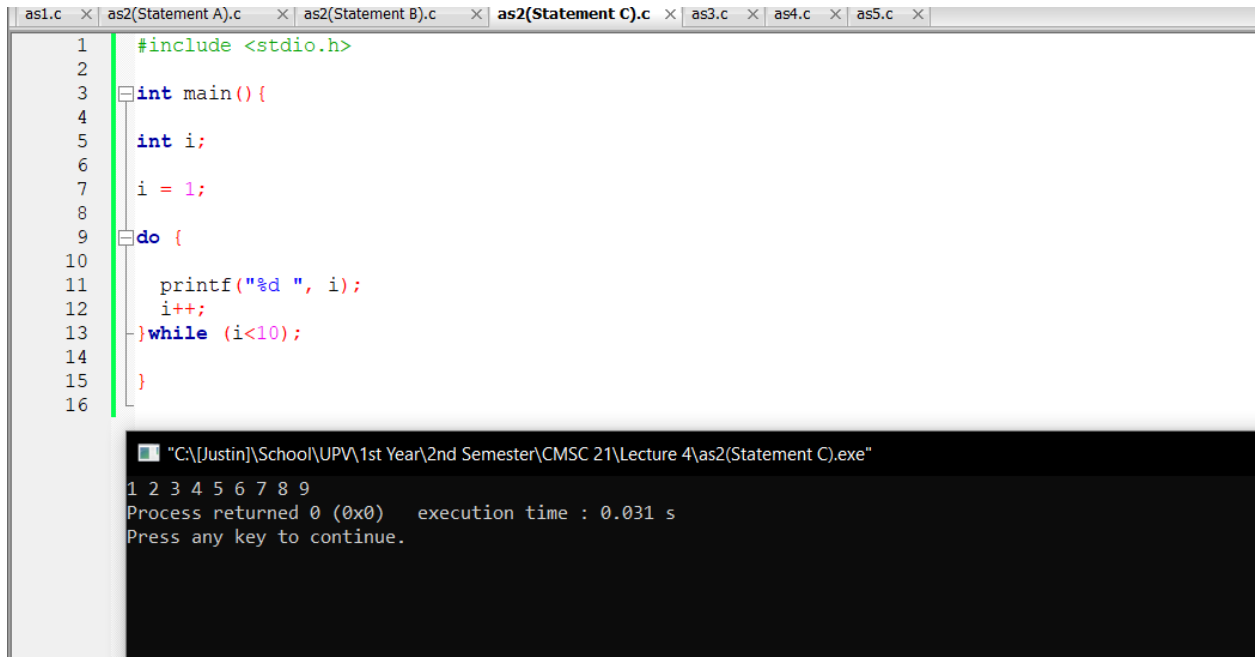
The screenshot shows a C program in a text editor with the following code:

```
1 #include <stdio.h>
2
3 int main(){
4
5     int i;
6
7     for (i=1;i<10;i++){
8
9         printf("%d ", i);
10     }
11
12 }
13
14
```

Below the code is a terminal window titled "C:\[Justin]\School\UPV\1st Year\2nd Semester\CMSC 21\Lecture 4\as2(Statement B).exe". The terminal output is:

```
1 2 3 4 5 6 7 8 9
Process returned 0 (0x0)   execution time : 0.026 s
Press any key to continue.
```

### Statement C:



The screenshot shows a code editor with several tabs. The active tab is 'as2(Statement C).c'. The code in the editor is as follows:

```
1  #include <stdio.h>
2
3  int main(){
4
5      int i;
6
7      i = 1;
8
9      do {
10
11         printf("%d ", i);
12         i++;
13     }while (i<10);
14
15 }
16
```

Below the code editor is a black console window showing the output of the program:

```
"C:\[Justin]\School\UPV\1st Year\2nd Semester\CMSC 21\Lecture 4\as2(Statement C).exe"
1 2 3 4 5 6 7 8 9
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

Example code and output showing C as not equivalent to A and B.

### Statement A and B:



The screenshot shows a code editor with several tabs. The active tab is 'as2(Statement A).c'. The code in the editor is as follows:

```
1  #include <stdio.h>
2
3  int main(){
4
5      int i;
6
7      i = 10;
8
9      while (i<10){
10
11         printf("%d ", i);
12         i++;
13     }
14
15 }
16
```

Below the code editor is a black console window showing the output of the program:

```
"C:\[Justin]\School\UPV\1st Year\2nd Semester\CMSC 21\Lecture 4\as2(Statement A).exe"
Process returned 0 (0x0)   execution time : 0.027 s
Press any key to continue.
```

The screenshot shows a code editor with a file named `as2(Statement B).c`. The code is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4
5     int i;
6
7     for (i=10; i<10; i++){
8
9         printf("%d ", i);
10
11     }
12
13 }
14
```

Below the code editor is a terminal window titled `"C:\[Justin]\School\UPV\1st Year\2nd Semester\CMSC 21\Lecture 4\as2(Statement B).exe"`. It displays the output of the program:

```
Process returned 0 (0x0)   execution time : 0.026 s
Press any key to continue.
```

### Statement C:

The screenshot shows a code editor with a file named `as2(Statement C).c`. The code is as follows:

```
1 #include <stdio.h>
2
3 int main() {
4
5     int i;
6
7     i = 10;
8
9     do {
10
11         printf("%d ", i);
12         i++;
13     } while (i<10);
14
15 }
16
```

Below the code editor is a terminal window titled `"C:\[Justin]\School\UPV\1st Year\2nd Semester\CMSC 21\Lecture 4\as2(Statement C).exe"`. It displays the output of the program:

```
10
Process returned 0 (0x0)   execution time : 0.026 s
Press any key to continue.
```

3.

Code:

```
1  #include <stdio.h>
2
3  int main(void) {
4
5  int i;
6
7  for (i = 1; i <= 128; i *= 2){
8  printf("%d ", i);
9  }
10
11 return 0;
12
13 }
14
```

Sample output:

```
1  #include <stdio.h>
2
3  int main(void) {
4
5  int i;
6
7  for (i = 1; i <= 128; i *= 2){
8  printf("%d ", i);
9  }
10
11 return 0;
12
13 }
14
```

"C:\\Justin\\School\\UPV\\1st Year\\2nd Semester\\CMSC 21\\Lecture 4\\as3.exe"

1 2 4 8 16 32 64 128

Process returned 0 (0x0) execution time : 0.028 s


Press any key to continue.

#### 4. Code:

```
1  #include <stdio.h>
2
3  int main(){
4
5  int exponent,i,base;//Declares variables exponent, i, and base as integer types.
6
7  printf("Please enter an exponent of 2.\n");//Asks the user to enter an exponent of 2.
8  scanf("%i",&exponent);//Uses scanf function to receive input from the user and places it at address variable exponent.
9
10
11  i = 1;//i is equal to 1.
12  base = 1;//base is equal to 1.
13
14  if (exponent == 0){//if user inputs 0 as exponent input, program uses this if statement to print 2 raised to zero is equal to 1.
15
16      printf("2 raised to 0 is equal to 1.\n");
17  }
18
19  else if (exponent > 0){//else if the user inputs anything greater than 0, the program continues,
20
21      while (i <= exponent){//while i is less than or equal to exponent, the while loop performs the loop body. The while loop is created in a w
22
23          base *= 2;
24          i++;
25      }
26
27      printf("2 raised to %i is equal to %i.\n", exponent,base);//After the while loop, the program outputs the final result of 2 raised to what
28  }
29
30
31 }
```

#### Sample output:

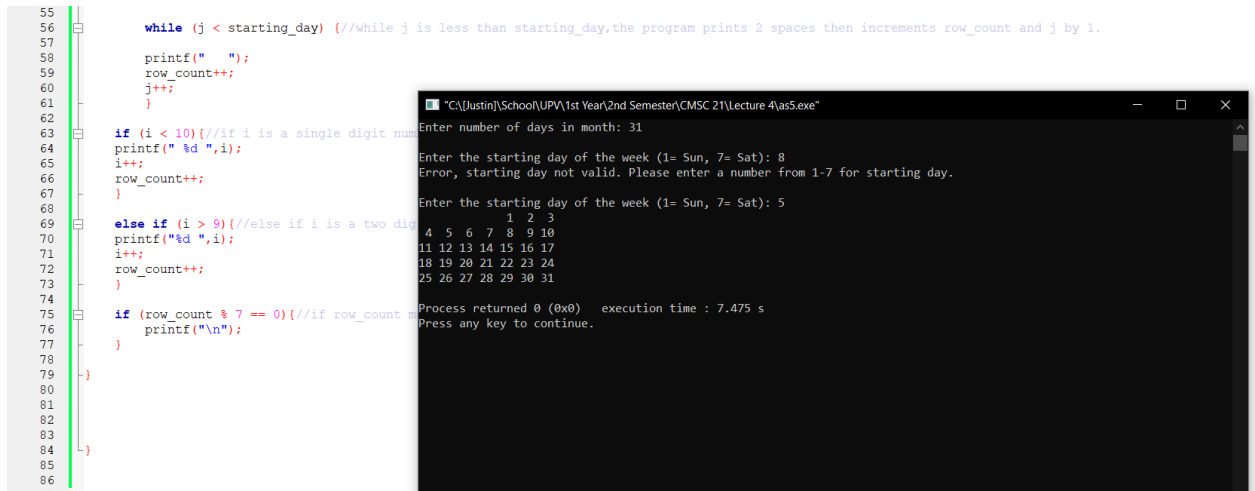
```
1  #include <stdio.h>
2
3  int main(){
4
5  int exponent,i,base;//Declares variables exponent, i, and base as integer types.
6
7  printf("Please enter an exponent of 2.\n");//Asks the user to enter an exponent of 2.
8  scanf("%i",&exponent);//Uses scanf function to receive input from the user and places it at address variable exponent.
9
10
11  i = 1;//i is equal to 1.
12  base = 1;//base is equal to 1.
13
14  if (exponent == 0){//if user inputs 0 as exponent input, program uses this if statement to print 2 raised to zero is equal to 1.
15
16      printf("2 raised to 0 is equal to 1.\n");
17  }
18
19  else if (exponent > 0){//else if the user inputs anything greater than 0, the program continues,
20
21      while (i <= exponent){//while i is less than or equal to exponent, the while loop performs the loop body. The while loop is created in i
22
23          base *= 2;
24          i++;
25      }
26
27      printf("2 raised to %i is equal to %i.\n", exponent,base);//After the while loop, the program outputs the final result of 2 raised to
28  }
29
30
31 }
```



## 5. Code:

```
1  #include <stdio.h>
2
3  int main() {
4
5      int days, starting_day, test, test1, i, j, row_count; //Declares days, starting_days, test, test1, i, j, and row_count as integer type variables.
6
7      //variables for the program.
8      test = 0; //used for 1st test of user input
9      test1 = 0; //used for 2nd test of user input.
10     i = 1; //used for printing the calendar.
11     j = 1; //used for printing the space as indicated by starting day.
12     row_count = 0; //used to print a new line every time row reaches 7 spaces.
13
14
15     while (test < 1) { //while loop for checking if user input is between 28-31. while test is less than 1, the loop body repeats.
16         printf("Enter number of days in month: "); //asks the user to input number of days in month.
17         scanf("%d", &days); //uses scanf function to receive input and place it into address variable days.
18
19         if (days > 31 || days < 28) { //if days is greater than 31 or less than 28, the program informs the user their input is not valid and will add 0 to test.
20
21             printf("Error, entered days not valid. Please enter 28-31 for days in month.\n");
22             test += 0;
23
24         }
25
26         else if (days > 27 && days < 32) { //else if days is between 28-31, then the program adds 1 to test so that the program can continue.
27             test++;
28         }
29     }
30
31
32
33     while (test1 < 1) { //while loop for checking if user input is between 1-7 for the starting day of the week. while test1 is less than 1, the loop body repeats.
34
35         printf("\nEnter the starting day of the week (1= Sun, 7= Sat): "); //asks the user to input the starting day of the week.
36         scanf("%d", &starting_day); //uses scanf function to receive input and places it at address variable starting_day.
37
38         if (starting_day > 7 || starting_day < 1) { //if starting day is less than 1 or greater than 7, the program informs the user that their input is not valid and will add 0 to test1.
39
40             printf("Error, starting day not valid. Please enter a number from 1-7 for starting day.\n");
41             test1 += 0;
42
43         }
44
45         else if (starting_day >= 1 && starting_day <= 7) { //else if starting_day is between 1-7, the program adds 1 to test1 so it can continue.
46             test1++;
47         }
48     }
49
50
51
52
53     while (i <= days) { //while i is less than or equal to days,
54
55         while (j < starting_day) { //while j is less than starting_day, the program prints 2 spaces then increments row_count and j by 1.
56
57             printf("  ");
58             row_count++;
59             j++;
60         }
61
62         if (i < 10) { //if i is a single digit number, it prints i with one space before and after it then increments i and row_count by 1.
63             printf(" %d ", i);
64             i++;
65             row_count++;
66         }
67
68         else if (i > 9) { //else if i is a two digit number, it prints i followed by only one space after it, then increments i and row_count by 1.
69             printf("%d ", i);
70             i++;
71             row_count++;
72         }
73
74         if (row_count % 7 == 0) { //if row_count modulo 7 is equal to 0, meaning that the spaces filling one row is 7, the program prints a new line.
75             printf("\n");
76         }
77     }
78
79 }
80
```

Sample output:



The image shows a code editor on the left and a terminal window on the right. The code editor displays C code for printing a calendar. The terminal window shows the program's execution, including prompts for the number of days in a month and the starting day of the week, followed by the printed calendar output.

```
55
56     while (j < starting_day) { //while j is less than starting_day, the program prints 2 spaces then increments row_count and j by 1.
57         printf("  ");
58         row_count++;
59         j++;
60     }
61
62
63     if (i < 10) { //if i is a single digit number
64         printf(" %d", i);
65         i++;
66         row_count++;
67     }
68
69     else if (i > 9) { //else if i is a two digit number
70         printf("%d ", i);
71         i++;
72         row_count++;
73     }
74
75     if (row_count % 7 == 0) { //if row_count is a multiple of 7, print a newline
76         printf("\n");
77     }
78 }
79
80
81
82
83
84
85
86
87
```

Terminal Output:

```
"C:\[Justin]\School\UPV\1st Year\2nd Semester\CMSC 21\Lecture 4\as5.exe"
Enter number of days in month: 31
Enter the starting day of the week (1= Sun, 7= Sat): 8
Error, starting day not valid. Please enter a number from 1-7 for starting day.
Enter the starting day of the week (1= Sun, 7= Sat): 5
   1  2  3
  4  5  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 31
Process returned 0 (0x0)   execution time : 7.475 s
Press any key to continue.
```

GitHub link: <https://github.com/zaxepaz/CMSC-21/tree/master/Lecture%204/Assignments>