Emmanuel Ian A. Albeza CMSC 21 Lec BS Computer Science - I

1. CODE

```c
#include <stdio.h>

int main (void)
{
    int i;

    i = 1;
    while (i <= 128) {
        printf("%d ", i);
        i *= 2;
    }

    return 0;
}
```

OUTPUT

```
1 2 4 8 16 32 64 128
```

2. The do-while loop is not equivalent to the other loops

CODE

```c
#include <stdio.h>

int main (void){

    int i = 10;

    printf("While loop output: \n");

    while (i < 10){
        printf("%d ", i);
        i++;
    }

    printf("\n");
    printf("For loop output: \n");

    for (i = 10; i < 10; i++){
        printf("%d ", i);
    }

    printf("\n");
    printf("Do-while loop output: \n");

    i = 10;

    do{
        printf("%d ", i);
        i++;
    }while (i < 10);
}
```

OUTPUT

```
While loop output:

For loop output:

Do-while loop output:
10
```

3. CODE

```c
#include <stdio.h>

int main (void)
{
    for (int i = 1; i <= 128; i *= 2)
        printf("%d ", i);

    return 0;
}
```

OUTPUT

```
1 2 4 8 16 32 64 128
```

4. CODE

```c
#include <stdio.h>

int main (void){

    int power, number;

    // Inputting of value o n
    printf("Please input a number: ");
    scanf("%d", &number);

    printf("n    2 to the n\n");
    printf("_    __________\n");

    // For loop to display the correct value of 2 to the nth value
    for (int i = 0; i <= number; i++){
        if (i == 0){
            power = 1;
        }
        else{
            power = power * 2;
        }
        printf("%d     %d\n", i, power);
    }
}
```

SAMPLE OUTPUT

```
Please input a number: 10
n    2 to the n

_    __________
0     1
1     2
2     4
3     8
4     16
5     32
6     64
7     128
8     256
9     512
10     1024
```

5. CODE

```c
#include <stdio.h>

int main (void){

    int days, start;

    // Do-while loop to validate proper number of days
    do{
    printf("Enter number of days in month: ");
    scanf("%d", &days);
    } while (days < 28 || days > 31);

    // Do-while loop to validate proper start of the calendar
    do{
    printf("Enter the starting day of the week (1=Sun, 7=Sat): ");
    scanf("%d", &start);
    } while (start < 1 || start > 7);

    printf("\n");

    // Printnig of space for the start of the calendar
    for (int i = 1; i < start; i++){
        printf("    ");
    }

    // Printing of the calendar
    for (int i = 1; i <= days; i++){
        printf("%2d ", i);
        // Checks to see if each day is the end of the week then prints to the next line if yes
        if (((start + i - 1) % 7) == 0)
            printf("\n");
    }
}
```

SAMPLE OUTPUT

```
Enter number of days in month: 31
Enter the starting day of the week (1=Sun, 7=Sat): 2

    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
```

6. A
```c
bool pathway[8] = {[0] = true, [2] = true};
```
B
```c
bool pathway[8] = {true, false, true};
```

OUTPUT
(BOTH ARE SAME)

```
pathway[0] is open
pathway[1] is close
pathway[2] is open
pathway[3] is close
pathway[4] is close
pathway[5] is close
pathway[6] is close
pathway[7] is close
```

7. CODE

```c
#include <stdio.h>

#define COLUMN 9
#define ROW 9

int main(void){

    // Declaration of variable for user input and the road network array/graph
    int station;

    int road_networks[ROW][COLUMN] = {
        {1, 1, 0, 0, 0, 1, 0, 0, 0},
        {1, 1, 1, 0, 0, 0, 0, 0, 0},
        {0, 1, 1, 0, 1, 1, 0, 0, 1},
        {0, 0, 0, 1, 1, 0, 0, 0, 0},
        {0, 0, 0, 1, 1, 0, 0, 0, 0},
        {1, 0, 1, 0, 0, 1, 0, 0, 0},
        {1, 0, 0, 1, 0, 0, 1, 0, 0},
        {0, 0, 0, 0, 0, 0, 0, 1, 1},
        {0, 0, 0, 0, 0, 0, 0, 1, 1}
    };

    // Printing of the adjacency matrix
    printf("    THE ADJACENCY MATRIX    \n");
    printf("   a  b  c  d  e  f  g  h  i \n");

    for (int i = 0; i < ROW; i++){
        printf("%c ", 'a' + i);
        for (int j = 0; j < COLUMN; j++){
            if (i == 2 || i == 3 || j == 2 || j == 3){
                printf("[%d]", road_networks[i][j]);
            }
            else{
                printf("%2d ", road_networks[i][j]);
            }
        }
        printf("\n");
    }
```

```c
    //Taking the user input
    printf("\nWhat station are you currently in? Please enter a valid station.\n");
    printf("A = 0, B = 1, C = 2, D = 3, E = 4, F = 5, G = 6, H = 7, I = 8):\n");
    scanf("%d", &station);

    // Switch case to print the nearest charging station
    switch(station){
        case 0:
            printf("\nAt point: A \nPoint: C arrived at charging station.");
            break;
        case 1:
            printf("\nAt point: B \nPoint: C arrived at charging station.");
            break;
        case 2:
            printf("\nAt point: C \nPoint: C is a charging station.");
            break;
        case 3:
            printf("\nAt point: D \nPoint: D is a charging station.");
            break;
        case 4:
            printf("\nAt point: E \nPoint: D arrived at charging station.");
            break;
        case 5:
            printf("\nAt point: F \nPoint: C arrived at charging station.");
            break;
        case 6:
            printf("\nAt point: G \nPoint: D arrived at charging station.");
            break;
        case 7:
            printf("\nAt point: H \nNo charging station nearby.");
            break;
        case 8:
            printf("\nAt point: I \nNo charging station nearby.");
            break;
        default:
            printf("\nPlease enter a valid station.\n");
            main();
    }
}
```

SAMPLE OUTPUTS (testing the points)

```
    THE ADJACENCY MATRIX
   a  b  c  d  e  f  g  h  i
a  1  1 [0][0] 0  1  0  0  0
b  1  1 [1][0] 0  0  0  0  0
c [0][1][1][0][1][1][0][0][1]
d [0][0][0][1][1][0][0][0][0]
e  0  0 [0][1] 1  0  0  0  0
f  1  0 [1][0] 0  1  0  0  0
g  1  0 [0][1] 0  0  1  0  0
h  0  0 [0][0] 0  0  0  1  1
i  0  0 [0][0] 0  0  0  1  1

What station are you currently in? Please enter a valid station.
A = 0, B = 1, C = 2, D = 3, E = 4, F = 5, G = 6, H = 7, I = 8):
0

At point: A
Point: C arrived at charging station.
```

```
    THE ADJACENCY MATRIX
   a  b  c  d  e  f  g  h  i
a  1  1 [0][0] 0  1  0  0  0
b  1  1 [1][0] 0  0  0  0  0
c [0][1][1][0][1][1][0][0][1]
d [0][0][0][1][1][0][0][0][0]
e  0  0 [0][1] 1  0  0  0  0
f  1  0 [1][0] 0  1  0  0  0
g  1  0 [0][1] 0  0  1  0  0
h  0  0 [0][0] 0  0  0  1  1
i  0  0 [0][0] 0  0  0  1  1

What station are you currently in? Please enter a valid station.
A = 0, B = 1, C = 2, D = 3, E = 4, F = 5, G = 6, H = 7, I = 8):
3

At point: D
Point: D is a charging station.
```

```
    THE ADJACENCY MATRIX
   a  b  c  d  e  f  g  h  i
a  1  1 [0][0] 0  1  0  0  0
b  1  1 [1][0] 0  0  0  0  0
c [0][1][1][0][1][1][0][0][1]
d [0][0][0][1][1][0][0][0][0]
e  0  0 [0][1] 1  0  0  0  0
f  1  0 [1][0] 0  1  0  0  0
g  1  0 [0][1] 0  0  1  0  0
h  0  0 [0][0] 0  0  0  1  1
i  0  0 [0][0] 0  0  0  1  1

What station are you currently in? Please enter a valid station.
A = 0, B = 1, C = 2, D = 3, E = 4, F = 5, G = 6, H = 7, I = 8):
8

At point: I
No charging station nearby.
```

```
    THE ADJACENCY MATRIX
   a  b  c  d  e  f  g  h  i
a  1  1 [0][0] 0  1  0  0  0
b  1  1 [1][0] 0  0  0  0  0
c [0][1][1][0][1][1][0][0][1]
d [0][0][0][1][1][0][0][0][0]
e  0  0 [0][1] 1  0  0  0  0
f  1  0 [1][0] 0  1  0  0  0
g  1  0 [0][1] 0  0  1  0  0
h  0  0 [0][0] 0  0  0  1  1
i  0  0 [0][0] 0  0  0  1  1

What station are you currently in? Please enter a valid station.
A = 0, B = 1, C = 2, D = 3, E = 4, F = 5, G = 6, H = 7, I = 8):
13

Please enter a valid station.
    THE ADJACENCY MATRIX
   a  b  c  d  e  f  g  h  i
a  1  1 [0][0] 0  1  0  0  0
b  1  1 [1][0] 0  0  0  0  0
c [0][1][1][0][1][1][0][0][1]
d [0][0][0][1][1][0][0][0][0]
e  0  0 [0][1] 1  0  0  0  0
f  1  0 [1][0] 0  1  0  0  0
g  1  0 [0][1] 0  0  1  0  0
h  0  0 [0][0] 0  0  0  1  1
i  0  0 [0][0] 0  0  0  1  1

What station are you currently in? Please enter a valid station.
A = 0, B = 1, C = 2, D = 3, E = 4, F = 5, G = 6, H = 7, I = 8):
```