

Darah Via D. Moscoso

B.S. in Computer Science I

2022 – 08331

```
#include <stdio.h>

✓ int main(void){
    int i;
    i = 1;
    ✓ while (i <= 128) {
        printf("%d ", i);
        i *= 2;
    }
    return 0;
}

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

```

#include <stdio.h>

int main(void){
    int i;
    i = 10;
    while (i < 10) {
        printf("%d ", i);
        i++;
    }
    printf("\n");

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

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

```

2. c) do {...} while (i < 10);
the loop body of the first two statements will not be executed if the value of i is not satisfied, only the do - while.

```

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

```

- 3.

```

#include <stdio.h>
int main(void){
    long long int n, power2, c;
    printf("enter n: ");          //prompt the user to input n
    scanf("%lld", &n);           // read input and store to variable n
    power2 = 1;
    // loop n times to calcualte the power of 2
    for (c = 0; c < n; c++){
        power2 *= 2;
    }
    printf("%d = %lld", n, power2); //print the result
}

```

4.

```

1+ #include <stdio.h>
2
3 #define MAX_DAY 7
4
5 int main(void){
6     int numDays, startDay, days, count;
7     do{
8         printf("Enter number of days in month: ");
9         scanf("%d", &numDays);
10        printf("Enter the starting day of the week (1 = Sun, 7 = Sat):");
11        scanf("%d", &startDay);
12    }while(numDays < 28 || numDays > 31 || startDay < 1 || startDay > 7);
13
14    for (count = 1; count < startDay; count++){
15        printf(" ");          // print spaces before beginning of day
16    }
17    for (days = 1; days <= numDays; days++, count++){
18        if (days < 10){
19            printf(" %d ", days); // print spaces to align single-digit days to double-digit days
20        }else{
21            printf("%d ", days); // prit double-digit days
22            if (count % MAX_DAY == 0){ // change row if reached 7th column
23                printf("\n");
24            }
25        }
26    }
27

```

5.

6. a. `bool pathway[8] = {[0] = true, [2] = true};`
b. `bool pathway[8] = {true, false, true};`

```

1 #include <stdio.h>
2
3 #define ROW 9
4 #define COLUMN 9
5 #define CHARGING_STATION_INDEX_C 2
6 #define CHARGING_STATION_INDEX_D 3
7
8 int main(void){
9     int location_input, current_point;
10    char labels[ROW] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I'}; // for columns and labels
11
12    int road_networks[ROW][COLUMN] = {{1, 1, 0, 0, 0, 1, 0, 0, 0},
13                                       {1, 1, 1, 0, 0, 0, 0, 0, 0},
14                                       {0, 1, 1, 0, 1, 1, 0, 0, 1},
15                                       {0, 0, 0, 1, 1, 0, 0, 0, 0},
16                                       {0, 0, 0, 1, 1, 0, 0, 0, 0},
17                                       {1, 0, 1, 0, 0, 1, 0, 0, 0},
18                                       {1, 0, 0, 1, 0, 0, 1, 0, 0},
19                                       {0, 0, 0, 0, 0, 0, 0, 1, 1},
20                                       {0, 0, 0, 0, 0, 0, 0, 1, 1}};
21
22    // print the adjacency matrix
23    int row, column;
24    printf("   A   B   [C] [D]   E   F   G   H   I\n");
25    for (row = 0; row < ROW; row++){
26
27        if (row == 2 || row == 3){
28            printf("[%c]", labels[row]); // display the labels
29        }else{
30            printf("%c ", labels[row]);
31        }
32
33        for (column = 0; column < COLUMN; column++){
34            printf("%5d", road_networks[row][column]); // print the individual array values
35        }printf("\n");
36    }
37
38
39    printf("Which point are you located? 0 - A, 1- B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H\n");
40    scanf("%d", &location_input); // read user input for location
41
42
43    for (current_point = 0; current_point < ROW; current_point++){
44        if (location_input == CHARGING_STATION_INDEX_C){ // if the location input is the charging station C
45            printf("C is a charging station."); break;
46        }
47        else if (location_input == CHARGING_STATION_INDEX_D){ // if the location input is the charging station D
48            printf("D is a charging station"); break;
49        }
50        else if (current_point == location_input) { // if the location input is not C or D
51            printf("at point: %c\n", labels[location_input]);
52            for (current_point; current_point < ROW; current_point++){ // iterate the row starting on the location row
53                if (road_networks[current_point][CHARGING_STATION_INDEX_C] == 1){ // if there is a road network to charging station C
54                    printf("point: C arrived to charging station"); break;
55                }
56                else if (road_networks[current_point][CHARGING_STATION_INDEX_D] == 1){ // if there is a road network to charging station D
57                    printf("point: D arrived to charging station"); break;
58                }
59                else{
60                    continue; //proceed to the next row if charging station is not found on the location's row
61                }
62            }
63        }
64    }
65 }
66

```