

1. What is the output of the following program?

```
#include <stdio.h>

int main(void)
{
    int i;

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

    return 0;
}
```

1 2 4 8 16 32 64 128

```
loops and arrays > C as1.c > main(void)
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 }
```

```
yvonne.lin@Yvannes-MacBook-Air
1 2 4 8 16 32 64 128
yvonne.lin@Yvannes-MacBook-Air
```

2. Which one of the following statements is not equivalent to the other two (assuming that the loop bodies are the same)?

a) while (i < 10) {...}

b) for (; i < 10;) {...}

c) do {...} while (i < 10);

C. since it will execute the body first before checking the condition (i < 10)

Save your code as as2.c

```
1  #include <stdio.h>
2
3  int main(void){
4
5      // A. while loop
6      int i = 0;
7      while (i < 10){
8          printf("%d ", i);
9          i++;
10     }
11
12     // B. for loop
13     for (; i < 10; ){
14         printf("%d ", i);
15         i++;
16     }
17
18     // C. do while
19     do{
20         printf ("%d ", i);
21         i++;
22     }
23     while (i < 10);
24     return 0;
25 }
```

```
yvonneline@Yvannes-MacBook-
0 1 2 3 4 5 6 7 8 9
yvonneline@Yvannes-MacBook-
```

Convert item 1 into an equivalent for statement. You can validate your answer by checking if the produced outputs by both the while and for statements are similar.

Save your code as as3.c

loops and arrays > C as3.c > main(void)

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

```
yvonne lin@Yvannes-MacBook-Air
1 2 4 8 16 32 64 128
yvonne lin@Yvannes-MacBook-Air
```

4. Write a code that computes for the power of two:

TABLE OF POWERS OF TWO

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

```
loops and arrays > C as4.c > main(void)
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void) {
5
6      //prompts user for input
7      int n;
8      printf("Enter value of n: ");
9      scanf("%d", &n);
10     printf("n\t2^n\n");
11
12     for (int i = 0; i <= n; i++) {
13         // uses pow function to calculate 2^n
14         int power = (int) pow(2, i);
15         printf("%d\t%d\n", i, power);
16     }
17     return 0;
18 }
19
```

```
Enter value of n: 4
n      2^n
0      1
1      2
2      4
3      8
4      16
yvonnelin@Yvonnnes-MacBook
```

5. Write a program that displays a one-month calendar.

```
Enter number of days in month: 31
Enter the starting day of the week (1=Sun, 7=Sat): 3

    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
```

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int days, start, i, j;
5
6      // prompts user to enter the day and validates it
7      do{
8          printf("\nEnter number of days in month: ");
9          scanf("%d", &days);
10         if (days <=27 || days >= 32)
11             printf ("Error: Invalid number of days! Please try again.\n");
12     } while (days <=27 || days >= 32);
13
14     // prompts user to enter the starting day and validates it
15     do{
16         printf("\nEnter starting day of the week (1=Sun, 7=Sat): ");
17         scanf("%d", &start);
18         if (start < 0 || start > 7)
19             printf ("Error: Invalid input! Please try again.\n");
20     } while (start < 0 || start > 7);
21
22     // prints calendar header
23     printf("\n\n  S  M  T  W  TH  F  S\n\n");
24
25     // prints spaces before the starting day
26     for (i = 1; i < start; i++) {
27         printf("    ");
28     }
29
30     // prints the days of the month
31     for (j = 1; j <= days; i++, j++) {
32         printf("%4d", j);
33
34         // checks if i is the last day of the week
35         if (i % 7 == 0)
36             printf("\n");
37     }
38     printf ("\n\n");
39     return 0;
40 }
41
```

```
Enter number of days in month: 28
Enter starting day of the week (1=Sun, 7=Sat): 5

    S  M  T  W  TH  F  S
    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
```

- a. Revise line 16 such that you use a designated initializer to set pathways 0 and 2 to true, and the rest will be false. Make the initializer as short as possible.
- b. Revise line 16 such that the initializer will be short as possible (without using a designated initializer)

```
#include <stdio.h>
#include <stdbool.h>

#define NUM_PATHWAYS ((int) (sizeof(pathway)/sizeof(pathway[0])))

int main(){
    bool pathway[8] = {[0] = true, [2] = true};

    for (int i = 0; i < NUM_PATHWAYS; i++){
        if (pathway[i]){
            printf("pathways[%d] is open \n", i);
        }else{
            printf("pathway[%d] is closed \n", i);
        }
    }
    return 0;
}
```

```
#include <stdio.h>
#include <stdbool.h>

#define NUM_PATHWAYS ((int) (sizeof(pathway)/sizeof(pathway[0])))

int main(){
    bool pathway[8] = {true, false, true};

    for (int i = 0; i < NUM_PATHWAYS; i++){
        if (pathway[i]){
            printf("pathways[%d] is open \n", i);
        }else{
            printf("pathway[%d] is closed \n", i);
        }
    }
    return 0;
}
```

7. A road network can be represented using graphs. Assuming we have points / stations a, b, c, d, e, f, g, and h, we can represent a direct path from a point to another point using arrows.

```
1 #include <stdio.h>
2 #define ROW 9
3 #define COL 9
4
5 // initialize matrix
6 int main() {
7     int road_networks[ROW][COL] = {
8         {1, 1, 0, 0, 0, 1, 0, 0, 0},
9         {1, 1, 1, 0, 0, 0, 0, 0, 0},
10        {0, 1, 1, 0, 1, 1, 0, 0, 1},
11        {0, 0, 0, 1, 1, 0, 0, 0, 0},
12        {0, 0, 0, 1, 1, 0, 0, 0, 0},
13        {1, 0, 1, 0, 0, 1, 0, 0, 0},
14        {1, 0, 0, 1, 0, 0, 1, 0, 0},
15        {0, 0, 0, 0, 0, 0, 0, 1, 1},
16        {0, 0, 0, 0, 0, 0, 0, 1, 1}
17    };
18
19    // print matrix and add brackets for c and d
20    printf("Adjacency matrix:\n");
21    for (int i = 0; i < ROW; i++) {
22        for (int j = 0; j < COL; j++) {
23            if (i == 2 || j == 2 || i == 3 || j == 3) {
24                printf("[%d] ", road_networks[i][j]);
25            } else {
26                printf("%3d ", road_networks[i][j]);
27            }
28        }
29        printf("\n");
30    }
31
32    // prompts user for origin point
33    int origin;
34    printf("Which point are you located? 0- A, 1 -B, 2-C...8 -I?\n");
35    scanf("%d", &origin);
36
37    printf("At point: %c\n", 'A' + origin);
38
39    // if origin point happens to be exactly at c and d, print it is a charging station
40    for (int i = 0; i < ROW; i++) {
41        if (origin == 2){
42            printf("C is a charging station.");
43            break;
44        }
45        else if (origin == 3){
46            printf("D is a charging station");
47            break;
48        }
49    }
50    // if it is at other points, check if there is a direct path to the charging station
51    else if (i == origin){
52        for (int j = i; j <= COL; j++){
53            if (road_networks[j][2]==1){
54                printf("point: C arrived to charging station");
55                break;
56            }
57            else if (road_networks[j][3]==1){
58                printf("point: D arrived to charging station");
59                break;
60            }
61            else {
62                continue;
63            }
64        }
65    }
```

OR

```
73 #define COL 9
74
75 // initialize matrix
76 int main() {
77     int road_networks[ROW][COL] = {
78         {1, 1, 0, 0, 0, 1, 0, 0, 0},
79         {1, 1, 1, 0, 0, 0, 0, 0, 0},
80         {0, 1, 1, 0, 1, 1, 0, 0, 1},
81         {0, 0, 0, 1, 1, 0, 0, 0, 0},
82         {0, 0, 0, 1, 1, 0, 0, 0, 0},
83         {1, 0, 1, 0, 0, 1, 0, 0, 0},
84         {1, 0, 0, 1, 0, 0, 1, 0, 0},
85         {0, 0, 0, 0, 0, 0, 0, 1, 1},
86         {0, 0, 0, 0, 0, 0, 0, 1, 1}
87     };
88
89     // print matrix and add brackets for c and d
90     printf("Adjacency matrix:\n");
91     for (int i = 0; i < ROW; i++) {
92         for (int j = 0; j < COL; j++) {
93             if (i == 2 || j == 2 || i == 3 || j == 3) {
94                 printf("[%d] ", road_networks[i][j]);
95             } else {
96                 printf("%3d ", road_networks[i][j]);
97             }
98         }
99         printf("\n");
100     }
101
102     int origin;
103     printf("Which point are you located? 0- A, 1 -B, 2-C...7 -H?\n");
104     scanf("%d", &origin);
105     switch(origin){
106         case 0: case 1: case 5:
107             printf("\nAt point: %c\npoint: C arrived to charging station", 'A' + origin);
108             break;
109         case 2:
110             printf("\nAt point: %c\npoint: C is a charging station", 'A' + origin);
111             break;
112         case 3:
113             printf("\n At point: D\npoint: D is a charging station", 'A' + origin);
114             break;
115         case 4: case 6:
116             printf("\nAt point: E\npoint: D arrived to charging station", 'A' + origin);
117             break;
118         case 7:
119             printf("\nAt point: H\npoint: No nearest charging station", 'A' + origin);
120             break;
121         default:
122             printf("Choose only from Stations 1-7");
123             break;
124     }
125     return 0;
126 }
```


